Haaga-Helia
University of Applied Sciences

Establishing a community for an open-source software startup

Anna Lekanova

Master's thesis
Degree Program in
Communication Management
2020

Haaga-Helia
University of Applied Sciences

Abstract

20.5.2020

Author(s)
Anna Lekanova

Degree programme
Communication Management

Report/thesis title	**Number of pages and appendix pages**
Establishing a community for an open-source software startup	**91 + 30**

For the open-source organisation community is a powerful tool. It helps to reach out to developers and other stakeholders vital for the growth of the project. The community enables public awareness, allows saving human resources and marketing costs along with bringing first adopters.
This study examined the possibilities of building a community for an open-source software project. It aims to find the most efficient form of the community for engaging the stakeholders to drive the daily development and recurrent improvement of an open-source project Fluence network.

The study describes the characteristics of the community for an open-source project. The theoretical framework for this study proposes considering the community in the context of the stakeholder communication concept. The research highlights the possible application of a variety of stakeholder communication models to the community building phenomena.

The data for this thesis was gathered from a set of stakeholder groups: the core teams of open-source projects as well as their environments. The core teams were interviewed through semi-structured focus group interviews, followed by one-on-one video call interviews. The online survey sampled the wider environment. The data collection methods are completed by analyzing the previous researches and observation.

The findings include the plan of the community for an open-source project; practical recommendations for establishing a community along with the detailed proposal of an online community platform, offline events plan followed by applied guidelines for each stage of the community development.

The key finding of the thesis clarified the most engaging efficient model for the community which includes diversifying tool base as follows: online platforms for daily conversations, wiki, code libraries, collaborative work, along with consistent offline meetings.

Keywords
Community, stakeholder communication, developer community, open source

Table of Contents

1 Introduction

This chapter introduces the basic framework of the study, the structure of the research, problem statement, research questions and objectives.

The great number of enthusiasts is one of the key success of open-source software projects. Enthusiasts develop the open source voluntarily because their personal values reflect to the values of open source. (Smith 2015, 33). Therefore, growth driver of open source is pure motivation along with inspiration additionally to the intellectual and creative work. This arrangement however has limitations for the reason that inspiration has always contextual nature, though is necessary for completing any project. Establishing a community of developers would be a solution to this problem provided that losing the motivation an active contributor could be substituted with one who feels motivated at that moment, which will have positive effect on the open-source software project in general. New members of open-source (OS) project community or open-source community (OS community) are not only audience but also helpers who could bring fresh ideas to the development of the code, valid points of view (Oram, Bhorat 2018, 29).

For the open-source organization community is a powerful tool. It is the way to reach out to developers, to test the prototype, to get feedback, to get contributors to the code, to spread the word and find first adopters.

Therefore, the community is an essential part for any project progress and its team, the core. The term “core” used in this work mean the team that started the project, created and elaborated the idea, concept, e.g. creators of the “initial product”.

Since open source is thriving and new OS projects are constantly arising, the research of the topic of stakeholders’ involvement is becoming more relevant. This research is targeted at designing the recommendations to establish a community of developers for launching the open-source software, or open-source project. The study includes the theoretical basis of stakeholder communication theory as well as concepts of an open source and its specific aspect, online community, developers of open source.

1.1 Case company, relevance of the study of establishing the developer community

This subchapter describes the case company for the study, its relevance for the market as well as the need for establishing a community.

The research is targeted to study the case which is an open-source project run by the core team, organization Fluence Labs. The company was established back in august 2017, the employees are located decentrally, in variety of the countries and cities. The main direction of the company activity is development of the decentralized approach to processing, storing and managing data, the platform for decentralized applications. In the framework of this understanding, a decentralized approach means that the traffic load is distributed across multiple servers. In fact, the company's product is a protocol “Fluence” for decentralized processing including database management, i.e. decentralized database management system, which could be functioned as a service for other build-on-top decentralized applications.

The relevance of product development is caused by a number of the following factors or disadvantages of using centralized database:

- High costs of storing big data in a centralized format;
- Uncertain security level;
- Data privacy abuse by government and tech giants;
- Data surveillance (NSA Surveillance, aclu.org);
- Upending the centralized data storage;
- The possibility of data corruption within the environment;
- The complexity of money transfers as well as their tracking;
- The ability to close access to data by The Big Five or so-called “FAAMG” known as tech giants having the biggest data ownership: Google, Amazon, Facebook, Apple and Microsoft (Manjoo, 2017).

These problems stimulate the emergence of the need to create such a platform, system or technology, which would be independent on third party, and would not be controlled from the outside.

Attempts to create an approach to storing and processing data in a decentralized form, in part outside the main chain, considering placing on the blockchain, were undertaken by a number of companies, in particular, the software developer companies Plasma and Bluzelle. However, the existing competitor implementation of the platform for decentralized applications has a number of problems, for example:

- Lack of secure access to data;
- Low platform efficiency due to low latency/speed;
- Lack of economical incentivization, etc.

Fluence product is targeted to solve these problems with through the solid network architecture. The Fluence network is built on Kademlia algorithm (an algorithm designed for decentralized networks), consisting of nodes combined in clusters of 3-25 nodes. Node (from lat. Nodus – 'knot') is any computer or device connected to the Fluence network. The nodes of the decentralized network are interacting via peer-to-peer protocol Fluence, for exchanging the data. Interaction with the data is based on the Merkle Tree, more precisely its modified version, which enables the node to check the results of queries without access to the data itself. Data on the network is replicated within the cluster, so that in the event of failure of one or more of the nodes, the availability and integrity of the data for the client is ensured. Thus, Fluence project solves the existing challenges of centralized internet:

- Decentralized data processing and storage is safer due to the fact of no trust (nodes do not need to trust a server or each other, because the chain of trust algorithm is built in the network);
- More effective processing due to the higher latency, because it responds to requests from the node that is geographically closest to the client;
- Development is more effective, since it takes less costs due to the location of nodes around the world;
- The platform can be used to build and/or deploy any applications to peer-to-peer, including those that have no relation to the blockchain.

The key concept of the project is to make sure that the data is processed and stored by the client/user without sending it through the centralized server. That way government or tech giants are not able to track or surveil the information flows, the data privacy and the right of freedom returned to the user.

Significantly, Fluence Labs is a core team that develops and runs the protocol; Fluence is a protocol that Fluence Labs is building. As the project is completely open source (the source code is stored in public repositories in GitHub), the aim of the Fluence Labs organisation is to build a community around the protocol, Fluence network, which draws the attention to the issue of peer-to-peer technology advantages. The community is needed to support the network, run the nodes in the network to process and store the. data; to build an ecosystem around the network of variety of applications with the same purpose to bring the users freedom of choice and the privacy right back. The aim of the

planned community is to attracts developers which would build the applications and services on top of the (or using) the Fluence protocol, support the core protocol, so the core team is not own any code as such, because the code is public, so that core developer team could be substituted by other developers from the community which reflect the same values of freedom and privacy advocacy, independence of the big tech.

The project of the company appears in the form of a startup, the main characteristics of which are:

- Limited, lean project funding;
- A small number of people in the team;
- Multitasking.

These factors impose the need for interaction between representatives of the Fluence project and the environment. Thus, the basis for the development of a project, its promotion on the market, the involvement of users in its creation and improvement is the communication of project representatives (the core of the company) with its stakeholders.

1.2 Research questions and objectives

This subsection describes the research questions and objectives.

Today, it is the community that determines the success of an open-source project on the market, therefore it is important for Fluence to develop a community that would help attract new participants, as well as motivate and encourage the existing ones to work effectively.

The research problem of the given thesis is twofold: the theoretical implications in addition to the practical implications are analyzed.

The theoretical relevance of the research lies, first of all, in the fact that there are no comprehensive studies on OS projects and their features. Moreover, there are no specific researches devoted to assessing the contribution of the external environment to the effectiveness of OS projects; most of them, on the contrary, are devoted to assessing the participation of third parties in OS projects (that is, the absolute opposite).

The second aspect of the theoretical significance is the consolidation of the existing interpretations of the concept of “community of OS projects”, differentiating them from social groups.

This work discovers an algorithm for the development of OS projects in comparison to the organizational life cycle theory and a correlation between two concepts, which allows to introduce knowledge in the field of organizational development into community management.

The frequent replacement of the concept “open-source project participants (or members)” with the term “stakeholders” has become a driver for research on the stakeholder theory, as well as the possibilities of its application in managing developer communities. The result of the analysis showed that this theory can find application in the direction under study. The lack of applied research on the community management in open-source projects pushed the author of this paper to study the profile of a typical representative of this field. It was concluded that previously a software developer is not interested in communicating with bigger community due to personal psychological and social characteristics that can be traced in the framework of the professional transformation theory, nowadays, a modern IT professional is a person who is open to communication, however mainly in the professional environment. The solution to the problem can be the establishment of a professional community for software developers, aimed not only at collaborative volunteer work, but also at solving the communication problem. Thus, many sources of different focus were required to cover the subject of this paper, as gaps were found in the existing knowledge of the establishing and management of open-source projects.

The practical relevance of the research highlights an empirical study of the target groups’ requests and recommendations on establishing a community for open-source projects. It identifies the preferred forms of communication between the two main groups, in particular the “core” of the project and the environment. Moreover, the practical significance of the research also lies in the fact that the analysis of the results of the empirical study acknowledge an effective form of communication for its further implementation in the developer community, as well as develop basic recommendations for and the structure of involving the environment into the project.

The research subject is the open-source project community, and the research object is the process of effective communication between the core of the project and the stakeholders.

Effective communication in the framework of this study denotes an interaction when its participants are not only outside observers but are effectively involved in the process of developing open source software.

The key goal of the thesis is to develop recommendations for establishing a community of developers for creating and running open-source software.

The main research question is:

Which type of community is the most suitable for engaging stakeholders to drive its daily development and recurrent improvement?

The main research objective is to identify the type of the community which would engage stakeholders to drive its daily development and recurrent improvement.

A number of sub-questions around the OS project community for this research arise:

- How do OS projects operate? What are the specifics of open source and what's behind it?
- How is environment relevant to the OS project and its community?
- How does community for an OS project look like?
- Could the stakeholder theory be applied to the OS project community (OS community)?
- What is the target audience for the OS project community?
- What are the stakeholder groups for the OS project community?
- What are the requests and preferences of the target groups?
- What is the most suitable form of communication between the groups?
- How to establish a community for an open-source project?

To address these issues, following research objectives have been set:

- To review the concept of open-source projects, reveal the history of the development in this field, as well as its features;
- To evaluate the significance of the environment in reference to the implementation of OS projects;
- To identify the concept of a community of OS project developers and an algorithm for its development;
- To consider the possibility of using the stakeholder theory in engaging participants in an OS project;
- To determine the target audience of the OS project and its needs;
- To identify stakeholders for OS projects, the degree of their influence, as well as to identify priority groups;
- To carry out an empirical study of the target groups' requests and views on establishing a community for OS projects;

- To analyze the data obtained from the empirical research;
- To identify the preferred form of communication between the target groups of OS projects;
- To choose an effective form of communication for its further implementation in the developer community;
- To develop guidelines to build the structure of community of/for the developers;
- To draw conclusions.

A more detailed description of the research, as well as the conclusions for each of the tasks will be described in the following chapters. The data to support research objectives will be collected through qualitative and quantitative research. As a result of this work, the author will develop a community structure for an open-source project, as well as propose the specific guidelines for its establishing. Significantly, the proposed structure and guidelines elaborated can be applied not only on the case company, but also on the similar open-source projects with the aim of involving the stakeholders and establishing a community.

1.3 Thesis structure

This subchapter presents the structure of the thesis.

Structurally, the thesis is presented by one theoretical chapter and three empirical chapters. The paper additionally contains a chapter devoted to the findings of the research.

The theoretical basis for the research is presented in chapter two. It describes the concept, history and features of OS projects, reveals the importance of the environment in the implementation of OS projects, discusses the concept of a developer community and an algorithm for its development, as well as the possibilities of using the stakeholder theory in involving participants in an OS project.

The empirical research methodology and a description of the empirical research methods are stated in chapter three.

The results of the study of various groups, as well as a generalization of these results are given in chapter four.

Chapter five describes the planned structure for the developer community establishment, and provides basic recommendations for its management.

In conclusion, chapter seven highlights the findings of the study along with the reliability and validity of the research. In addition, reflections on training and suggestions for further research are also provided in chapter seven.

2 Literature Review

This chapter highlights the theoretical foundations of the research, reveals the concepts of "open source" and "community for developers", additionally overviews their features. Theoretical part of the work attempts to apply the well-known theories to the elaborative research elements; the need and urgency of their use is discovered.

2.1 Open-source software development project

The subsection discovers the reasons for open source movement, key characteristics and requirements for the open-source software.

There are two independent movements in Free Software: Open Source Initiative (OSI) and GNU Free Software Foundation (FSF) led by Richard Stallman. Both movements have different interpretations of the terms "free software" and "open source" (Brasseur 2018, 17).

The movement targeted to opening the code began with the emergence of personal computers (PC) and the commercialization of software. This aspect became the origin of the hacker movement, which lasted until 1984. In spite of the fact that this direction has mainly negative connotation in most sources, it was the driver of dynamic development of software of numerous directions, became the basis for research of codes, their copying and reconstruction (Fogel 2005, 25).

The degradation of the hacker movement, the loss of its representatives, was a prerequisite for the development of the GNU Project, the emergence of the "Free Software Foundation (FSF)" public organization headed by R. Stallman. The leader proposed the concept of "Copyleft," under which programs and programmers were guaranteed the four types of freedom shown in the Figure 1 below.

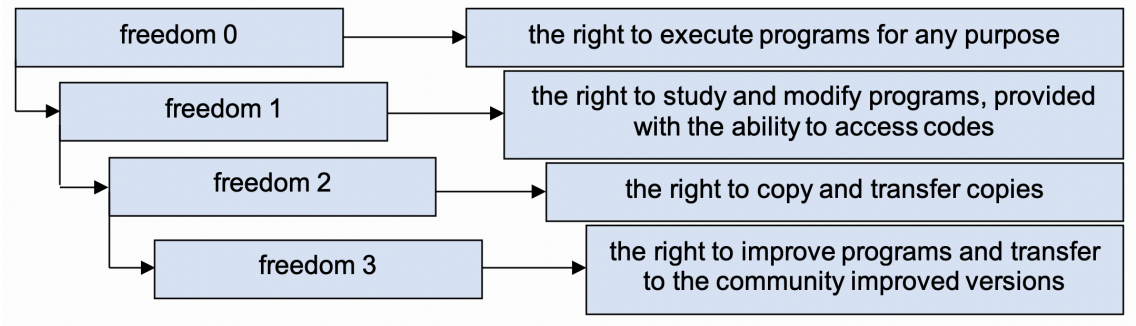


Figure 1. Types of freedom in the original concept of free software (adapted from the Stallman freedom model)

These rights are fixed in licenses of different types. This trend initiated many projects, servers and languages, including Linux, Send mail, Apache, My SQL, Python, Open Office, Mozilla and Debian (Haff 2018, 26).

The success of the trend, as well as its wide application, contributed to the emergence of another initiative, the Open Source Initiative, founded on 3 February 1998 in Palo Alto. Although based on the concept of "Free Software Foundation (FSF)", this is a completely different interpretation of classic hacker culture. The term Open Source was chosen to avoid the ambiguity of the English word free, which can be understood as both "free" derived from freedom and "free-of-charge" derived from gratuitous. This concept is related to the name of Eric Raymond. The difference from the altruistic idea (refusal of financial incentives principles) of free software resides in the fact that Open source was a certain business concept. The basis of the Open Source Initiative became to drive a wider public attention in free software and to involve large organizations in it. The philosophy of the new movement was described in Raymond's software article "The Cathedraland the Bazaar" (Raymond 2001, 32).

Today, open source is narrowly defined as applications whose code is publicly accessible i.e. open. In addition, open-source projects are openly distributed open-source software (Peatfield 2015, 19). Consequently, the number of researchers argue the key characteristics of the open-source project is open-source code. Researches specify other key elements of an open source:

- User interface (UI);
- Visual design;
- User experience, for example, actions of the user, usability;
- Media material such as audio, graphics, video, depending on the nature of the project;
- Project documentation, for instance, texts, guidelines, guidebooks, translations;
- Marketing;
- Legal part (Lindberg 2008, 41).

Fundamentally, there is a license at the heart of the open-source project which

The basis of an open-source project is a license which is an agreement for using and restriction of the use of software.

The basic requirements for open-source licenses in the Open Source Initiative are illustrated in the Figure 2.

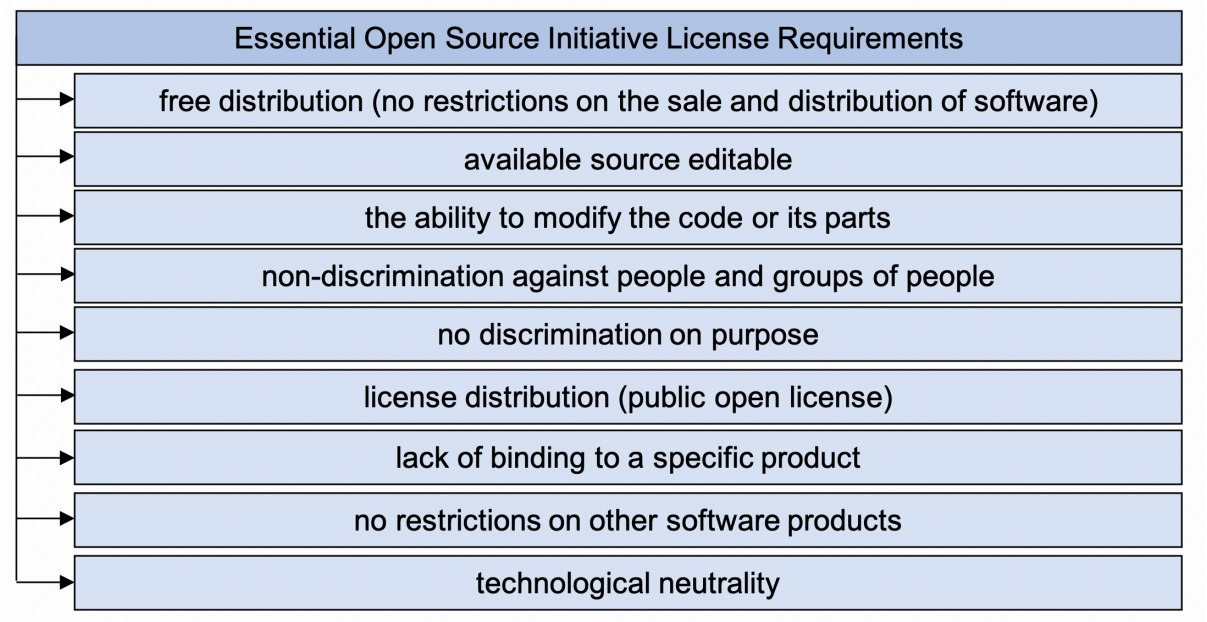


Figure 2. Key requirements for the open source software licenses by Open Source Initiative (adapted from Pazdera 2015, 8)

These features represent the characteristics of the open-source projects, namely free distribution, ability to edit the code or parts of the code, ability to fork the repository, human, product and technological neutrality.

Despite of the fact that any software is ultimately designed for the end user (developers could be end users as well), the special aspect of the open-source projects resides in the fact that any of the participants could contribute to the code, to participate in the development. Following that logic, any open-source project depends on motivation of the developers (or “geeks”, as they sometimes are called in the professional slang), the activists who continuously improve the code.

2.2 The emergency of a developer community for an open-source project

Open source projects are gaining momentum every day, new ones appear and popular projects such as Bootstrap, AngularJS, Elasticsearch, Symfony Framework, Swift, etc. are actively developing. Projects are drawing an increasing number of new developers, which in general offers an opportunity for enormous growth: projects and products are being improved, fixed, modernized, finalized, expanded, etc. Participants involved in a project are able to find many roles for themselves, and development in the following areas:

1. Testing of intermediate versions and releases. In any, even commercial projects, the development of the program functionality prevails over its testing. For this reason, normally before adding changes to the main branch of the repository, a call for testing is announced, i.e. engaging people who are ready to test the preliminary version. These tests are vitally important for checking the cross-platform operation of the software and let developers save time on testing, therefore to introduce the product to the market in a shorter time (Oram, Bhorat 2018, 54).

2. Writing, updating and translating the documentation. One of the key challenges for OS projects is neglectful document management. As a rule, a project is run by people interested in it, who are already versed in its current state without additional documentation. This dirves the necessity to involve participants in helping with enhancing the documentation. More often than not the documentation is either absent or exists in a poor state which it is quite difficult for a third-party reader to understand. There is a common problem of a documentation backlog in a dynamically growing project. All this affects the need for outsiders who are able to determine the gaps in the available documentation (Smith 2015, 89).

3. Search and correction of errors. One of the important stages of code development is testing, searching and fixing bugs. It is believed that this direction is the least challenging entry point for the “beginners” in open source. This type of participation can be manifested as writing the issue (bug registration), fixing it independently or monitoring its testing on users (fixing bugs, writing pull requests). Any program behaves differently depending on the version of the browser or the sequence of actions and testing it on different users. Maintaining a registry of errors will greatly simplify the development of the product. Another relevant advice is developing tests, which would determine the areas not involved in code testing deploy them (Brasseur 2018, 47).

4. Writing code examples and demo applications. In this area, stakeholders develop demo applications that show the use of the developed library or framework, aimed at introducing others to the project functionality. The main goal here is to show the functionality of the product to the user, without engaging the user in reading the documentation (Bellini 2019, 21).

5. Writing a new code. As part of an open source project, it is possible to propose a new code in accordance with the current state of the project. This also includes correcting a problem that occurs in the code, as well as adding missing functions, developing the fork version (Haff 2018, 84).

6. Design development. Most of project developers are involved only in the implementation of the technical design, which poses the problem of open-source projects having a poor visual design. Crucial that it is precisely the usable design (UI and UX) often attracts new users to the product, makes the onboarding easier, as well as draws stakeholders to its deploy (Lindberg 2008, 61).

7. Blogging with the aim of helping other users. It is also relevant to keep records on the implementation of the project publicly accessible, which consists in both the visibility of open source projects and identifying problems in the product development and a method for solving them. This direction allows to maintain interest and attention to the project, to create a knowledge base for those who will join it later (Volpi 2019, 73).

Thus, there are quite a few areas for the involvement of participants, who are an integral part in OS product development. Therefore, there is a need emerging to create a group of project participants, which should be a community of developers, namely like-minded people with the common goal to develop a software will not only satisfy the needs of the market, but will also be relevant and necessary in the stakeholders' applied work. On this basis, the establishment of a community of developers is crucial for increasing the effectiveness of OS projects. According to some researchers, products, projects and technologies are just a small part of the open source software movement. The community is the first and most important component (Block 2018, 112).

2.3 The development of algorithm of an open-source developer community

This subchapter argues the definition of an open-source community or the community of an open-source project as well as stages of participants (members or users) engagement in the open-source project.

First of all, it is worth mentioning that the community should be understood as groups of individuals with shared interests (Bacon 2018, 7). Community is vital for any open-source project, it is the "heart" of the project. Some researchers in this field emphasize that the main essence of an open-source project is not only programming but also socialization, creating a community (or association) of developers and stakeholders drawn by the project. If there are no discussions about a product, there is no conversations and interest from the community, core developers lose motivation for building the product (Effenberge 2018, 15).

Based on this, most OS projects have their own community such as places where developers and activists communicate, regardless of their background. It should be mentioned that the audience or clients are not a community for the reason that they do not have common goals. Despite the similarity of the terms "community" and "social group", they are not equal. A community is a voluntary association of people who are connected by mutual contacts and have a common goals, which they also voluntarily achieve without any material incentives. In turn, a social group is a small group of people united by common social activity, whose members are in direct contact and have emotional relations between each other. In general, social groups are people who are united by shared features (work, common interests, etc.), while communities are people who have common goal and connections and voluntarily participate in some activity (project, process, etc.). The main feature of a community is a dense network of connections within the participants. If an audience that does not have a connection with each other, they do not know each other at all and are only connected by the fact that they simply belong to the same group in social media is not a community, but a social group. Community is based on connections between people (Hintjens 2016, 48).

There are two types of communities distinctive for OS projects:

- Open communities – communities that are accessed by all participants, where any user can submit a proposal, point out a bug, modify code, etc.;
- Closed communities – communities that have invite-only entry or private, where every participant is thoroughly checked upon entry (Lindberg 2008, 34).

Both forms of community have the right to exist and apply according to the specifics of an OS project. At the same time, researchers emphasize that a closed community form has a greater impact on the quality of participants, their loyalty to the "core" of the project. Both closed and open projects have their own user communities, most of which are relatively passive in terms of their interaction with the rest of the community. On the other hand, any type of community may include members who choose to be more active, for example, by sending bug reports, helping other users, writing documentation, or advocating. Granting the access or strengthening of control over a project is often used as a reward for active participants in open-source projects (Masson 2005, 21).

Any community in its nascence goes through a series of stages, growing from a "core" of developers, usually numbering 2-5 people to a large-scale phenomenon. This process may take a long time (from 1 month to several years), which directly depends on the development of the original version (concept, idea, primary version, etc.) by the developers.

E. Raymond's proposes a notable approach, he says that a driver of establishing a community for an OS project is "something that works and can be tested, something to play with". He argues that "working" means not merely a finished product, but also something that can be refined and tested. The motto of open-source projects in the author's understanding should be "release frequent and early", which is explained by the fact that publishing the project before it is ready allows getting more early feedback from the market as well as increasing the reliability based on the early feedback. Thus, the starting point for community building is a "draft version of the product" that has the ability to be tested (Raymond 2001, 23).

In the initial phase, establishing the community is an attempt to see as many commonalities among its members as possible (their goals, roles, interests, etc.) and to standardize certain characteristics. This stage is organized by the principle of marketing/business intelligence. Once the participants and their interests have been identified, the stage of community building begins, where the goals and objectives of the project are set, including strategies for involving users in its development (Tamburri 2018, 25). The engagement of participants or users in the development of the OS project could take place in the following areas, as shown in Figure 3.

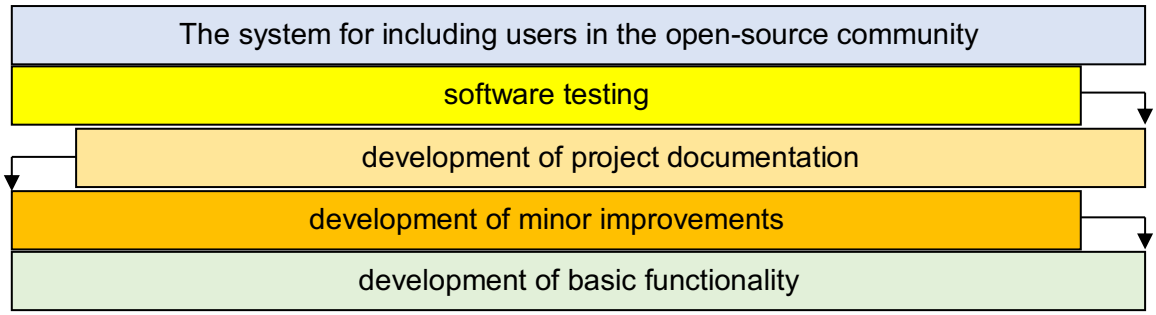


Figure 3. System of participants engagement in the open-source community (adopted from Tamburri)

Thus, the inclusion of participants is deployed according to the degree of qualification: from easy functionality in the testing of the product to the development of functions. The stage of participants' inspiration or stakeholders' motivation comes after defining the needs and the roles in product development. One of the most crucial areas here is the personal communication of developers or their representatives with users, as well as the initiation of newsworthy events to involve them in the conversation. An important integration point here is to create a platform for communication within social groups, GitHub, or own website. The online environment is the primary area of community building. It may bring together participants who are geographically distant from each other. One of the key areas is the organization of communication, as well as encouraging users to get to know each other, interact, comment, share problems, offer topics, and as

a result, actively participate in the development of an OS project. If a close network of understanding and communication among participants is established, the OS project may change its level of evolvement towards dynamic growth; it may obtain real supporters and loyal followers (Pearce 2018, 39).

Stimulating user activity and user desire to "talk about the project" may draw new participants. At this stage, it is critical to analyze the statistics of activity and the entry of participants, to choose what engages them and what repels them. Feedback is important, as the use of spontaneous surveys of participants is relevant, as well as the development of plans to increase the number of “high-quality” participants and their activity (Millington 2012, 89).

The main challenge is to organize support for participants so that their leaving is within the minimum threshold. The active decrease of the community size has a very negative impact on the motivation of the developers. Any input from participants in the community-building phase should be actively encouraged (Kraut, Resnick 2012, 78).

There is a fast growth in the community after the majority of participants are involved. New participants join mostly by invitation. Ignoring the numbers of growth with priority on the quality of participants is a key. This is connected with engaging a large number of participants who are less knowledgeable about the particularities of the project. It may become a stumbling point in the existing community, which may increase the conflict of the entire system, moreover cause an outflow of "minds and knowledge" (Holland 2015, 74).

The next step is to stabilize the community. At this stage, the growth of the number of participants is expected and the roles of each of them are formed. A feature of this stage is also a change in the authoritarian form of community governance towards consensus-based democracy. This factor is explained by the fact that people in the community become responsible for decision-making. Decisions, including minor ones, are made by an active community (which, as a general rule, does not exceed 1% of all participants). The rest are made according to the principle of "lazy consensus" when silence equals consent. A typical form of community development during this period is numerous voting. During this phase, care for new participants is often reduced, which can harm the further growth of the community. The degree of regulation is also growing; there are internal codes of communication, joint holidays, important dates. Standardization of the agreements allows giving the community its own life, which is reflected in the elaboration of the open-source product (Gamalielsson 2014, 134).

The danger of this stage is aligned with a risk of competitors ("fork"), which may lead to a decrease in user confidence. The fork might be a positive issue if the GitHub repository with the source code is forked i.e. duplicated by anyone else to be able to build on top of it with relation to the key source code. On the other hand, a fork could be negative if one who forks does not want to be related to the initial project and develops own ideas contradicting the main concept. The cause of a negative fork may be a technical, economic, or social problem that has emerged in the community and was not addressed promptly. Drivers of the fork could arise from negative statements about developers in the community, the insufficient system of filtering the potential community leads, or poor performance of community managers to integrate, activate, and motivate people. A fork within a community is a split into two camps: pioneers or immediate supporters of the original project and followers. Since communities are the driving force of any OS project, this problem is reflected directly in the development of the code itself. Participants express less interest in it, errors are not corrected, the motivation of the developers themselves decreases. There are certain technologies to avoid the negative fork (the main ones were described above, for instance, control of statements, motivation, the inclusion of "low-quality" participants into the community. Teambuilding techniques, e.g. joint events for participants, visiting industry events, etc., which take place in the offline environment, are used to overcome this. Also, some researchers emphasize that the inclusion of offline direction right from the beginning of community establishment, as well as its development by the type of "membership club" significantly reduces the possibility of group separation (Brasseur 2018, 76).

All in all, analyzing the algorithm stages of establishing and developing a community of engineers it can be summed up that community of the open-source project is progressing according to the standard product lifecycle curve (Iriberri, Leroy, 2009).

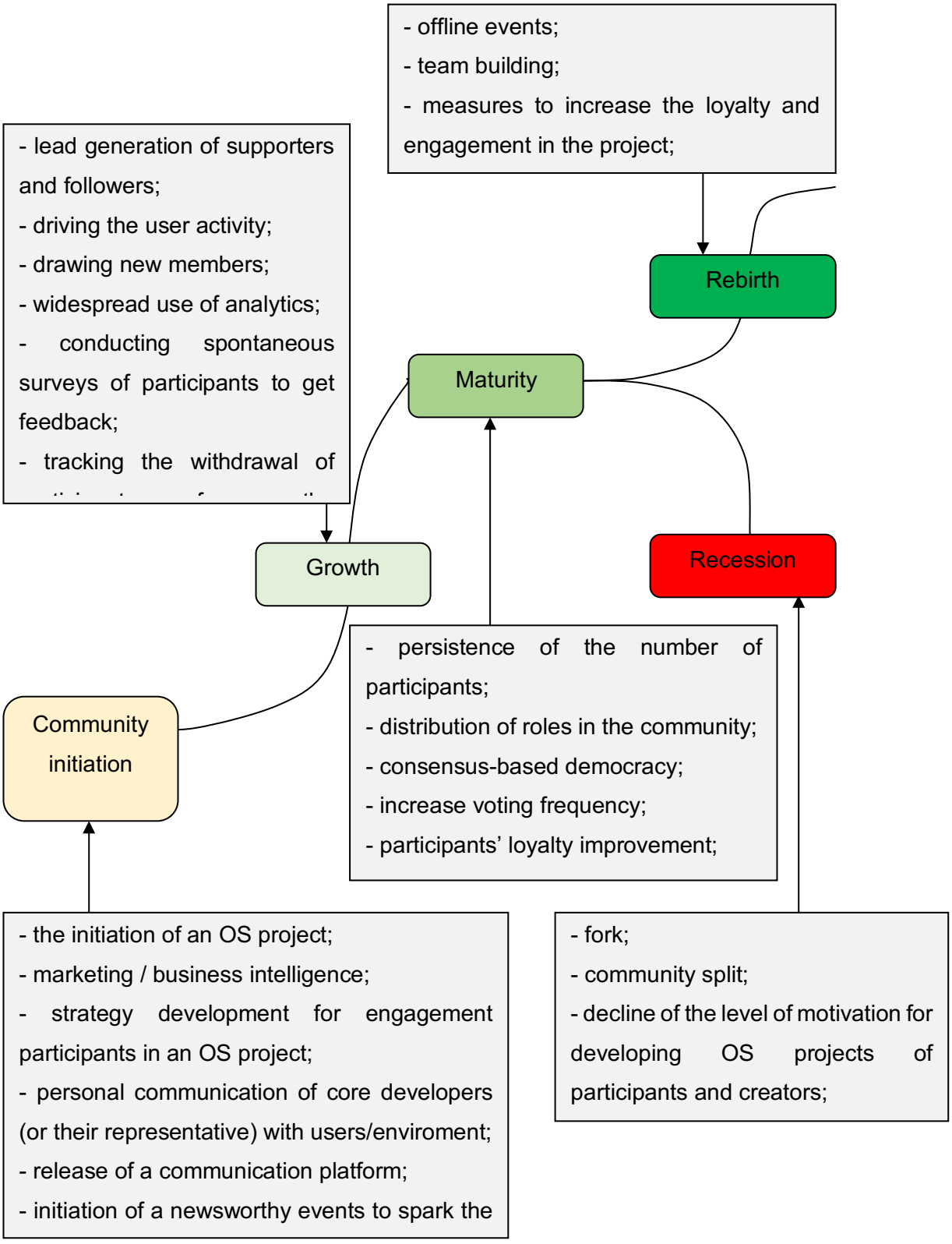


Figure 4. Development of the OS project community according to the product lifecycle

The author of the research drew this curve according to the specifics of each stage of the OS project. The efficient management of the community, tracking all stages of its development enables the OS project to develop itself. The growing community not only

nurtures faster release of new versions of applications but also helps to find bugs in it. The dynamic growth of participants has a positive impact on the motivations of volunteers working for the benefit of the product so a bigger number of contributors are involved in its development. When the decline stage occurs the level of motivation for software development by the core of the project decreases, and as a result, the product may be abandoned (first postponed, and then left out) (Clayton 2018, 59). Thus, aligning with the community life cycle management might be critical. Unfortunately, this area is not developed in the previous research enough, in particular applying the theory of the product or organization lifecycle to the OS community management.

Establishing a community around a startup or an OS project is a lean way to introduce the product to the market. A full-fledged community brings more benefits than a single one-off advertising campaign. However, despite the efficacy of this tool, a few companies are ready to publish their solutions, for example, in GitHub. It could be explained by the fact that organisations are convinced they open the project source code they might lose a competitive advantage, for instance, uniqueness, or open themselves to be copied or forked by competitors. This might explain the minority of open-source projects compared to proprietary ones (DiBona, Stone, Cooper 2008, 34).

In summary, it is critical to note that a process of building a community around an OS project is complicated moreover it might be slow-paced, takes lots of human resource time, and its success depends on a number of criteria. However, it is even more complicated for an OS project to grow without a community. Community building does not happen intrinsically and must be managed. All communities start with users drawn into the project by structured planning and daily activities. As soon as community members arrive, the path of meeting their expectations begins. A thriving developer community should meet and expand users' expectations but only provided with the provision of a community leader or community manager to keep it together and make sure that members are fully supported if they build on top of the projects’ OS software. In the long run, communities should have an open development structure to make sure that if key participants, including the founders, leave, the community could easily replace them, so open source continues to live.

2.4 Applying the stakeholder theory to the members engagement in the OS project

This subsection discusses the variety of theoretical stakeholder concepts and attempts applying them to the community for an open-source project.

In determining the participants, their preferences in the framework of marketing intelligence conducted at the first stage of the emergence of the community, there is no certain methodology for their study, rather this stage is presented as creative and meta-skill of a community manager. The specifics of involvement and participants' relevancy assessment in OS communities are somehow similar to stakeholder theory. Within the framework of this theory, experts emphasize that stakeholders have an impact on substantially all aspects of the modern enterprise. Unfortunately, there is no practice of involving this theory in OS projects, however since most projects of this type are carried out by stakeholders, the application of stakeholder theory in this area seems relevant. According to Cornelissen (2014), stakeholders are seen as groups, organizations, or individuals by which the company is influenced and on which organisation depends. Another definition of stakeholders is illustrating them as involved parties, individuals, or organizations that have rights, shares, claims, or interests in the organization or its elements that meet their needs and expectations (Smelt, Staves 2019, 17).

According to public statistics, the number of failed projects in Europe over the past ten years is about 25%; over the years this figure drops to 19% and then goes up again. Analyzing the statistics, experts emphasize that the low efficiency of project activities tends to be due to the lack of communication between project participants and stakeholders. The same setting could follow OS projects. Despite the lack of a certain number applied to specifically open source areas, eventually, it is the stakeholders who affect the success of startups (Smelt, Staves 2019, 9).

Further in the work, the stakeholder theory should be discussed to give a better introduction to the context of the research. The stakeholder theory originated by Edward Freeman in the 80s. He described the stakeholders as groups which affect or could be influenced by the organisation or its aims. Freeman argues that the companies are determined by the relationships with the stakeholders (Freeman, 2010, 46). Later the theory was developed by the number of research groups. The most recognized currently was led by Cornelissen arguing an organisation is more than just a profit-making tool but also an element of the environment in which it operates. A system is influenced by its environment, for instance, local communities, consumers, suppliers, public organizations, employees, investors, and shareholders. According to Cornelissen, managers should not make decisions that would limit the scope of choice of new generations in the future. Considering the organization to be an open system, the researcher adopted the belief that social issues in the organisation could be overcome if key institutions are rebuilt and stakeholders in the system interact effectively. As a result, the forthcoming of stakeholder theory has taken place and is now recognized worldwide. (Cornelissen 2014, 47).

The OS project is an initially open system, which depends on the environment, i.e. the direction fully satisfies all the selected characteristics of the theory (Freeman, Harrison, Barney, & Phillips 2018, 15). The OS project may be successful only if the environment takes part in its development, for instance, testing of intermediate versions, developing documentation and design, searching, and fixing bugs (Martin 2015, 44). Therefore, the theory of stakeholder engagement is relevant for application in this area.

There are several models for dividing stakeholders into groups in the practice of strategic management. The leading is an approach proposed by Newbould and Luffman (Stakeholder interests), Mitchell's model (Stakeholder Salience), Mendelow's model (Power-interest Matrix), balance, and network models. The authors Newbould and Luffman (1989) in their classic approach originated in 1979 argue the concept of dividing stakeholders into four main categories; each one is attributed to several incentives of participating in the organisation (Schmitz, Baum, Huett, & Kabbst 2019):

- Advocacy group which is related to funding the organisation, such as investors, business angels, venture funds, shareholders, banks, and others;
- The managers running the organisation;
- Employees working for the organisation;
- Other economy-related partners, for instance, suppliers, buyers, and other economic entities (Dobni, Luffman 2003).

The model is based on the steadiness of behavior and interests over time (Baugh 2015, 32). This model could be used for the research of stakeholders in the OS project (in particular, investor and partner groups). But the approach proposed by the authors is quite extensive for practical implementation, so four other models are used in practice: the Mitchell model, the Mendelow model, the balance model, and the network model.

1. Mitchell's model (Mitchell, Agle, & Wood 1997). Within this salience model, stakeholders are considered by the importance of their relevant attributes or characteristics. Each group is assessing for the possession of these attributes, and as a result, it becomes possible to attribute them to one or another class of significance, whose elements have (or do not have) the same set of attributes. These classes can be ordered based on the importance of the set of attributes corresponding to them. The importance of each interested party is estimated as the importance of the class it belongs to (Marin, Mitchell, & Lee 2015). According to the Stakeholder salience model, legitimacy, power, and urgency of requirements can be used as three main attributes.

These attributes are not static for each stakeholder group and can be subject to change (Mitchell, Agle, & Wood 1997). According to the researchers' work, these attribute combinations result in seven groups, shown in Figure 5.

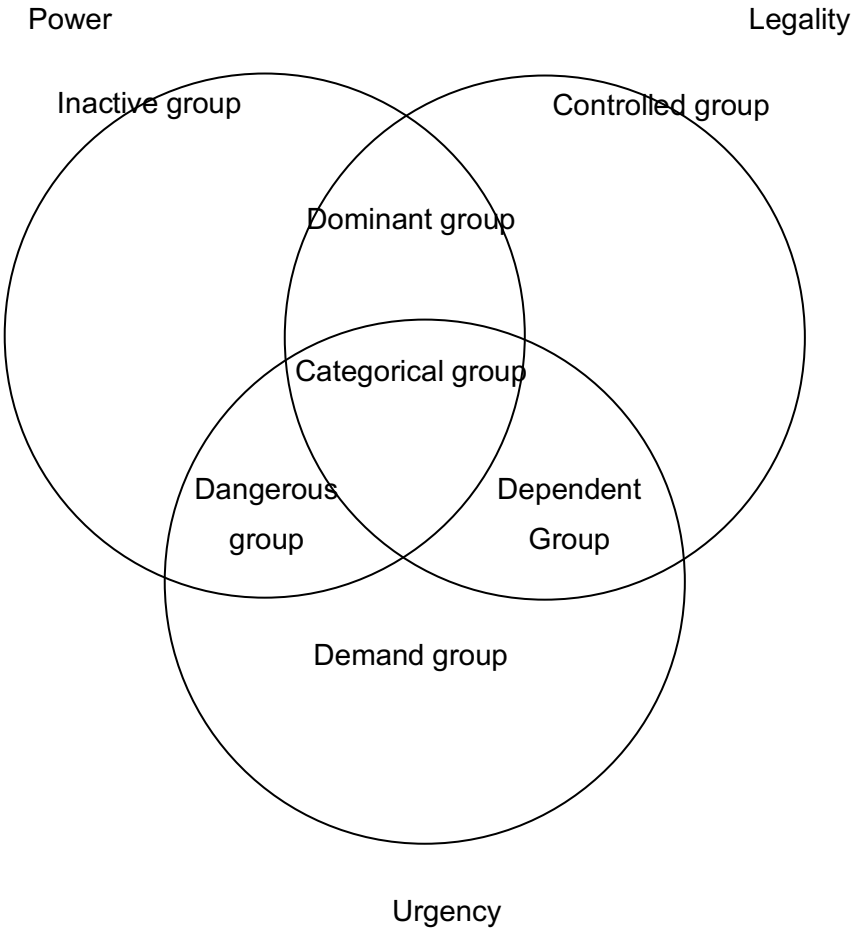


Figure 5. Stakeholder Salience model (adapted from Mitchell, Agle, & Wood 1997)

This study develops the application of the Stakeholder salience model to the classification of open-source stakeholders of the project is not optimal, as there is no key categorical group.

Mendelow's matrix (1991) is widely adopted by modern organisations. The model has become prototypical for a number of researches and has been adopted by Cornelissen (2012, 51). The matrix is based on the stakeholder ranking by two indicators:

- Power level (the ability to influence the organization);
- Interest level (the level of incentive to influence the organization).

This model is foremost often used to determine the influence of stakeholders on the setting of the organization's goals, as well as to analyze the conceivable conflicts among all stakeholders in achieving strategic goals (Cornelissen 2012).

The stakeholders' influence formula is at the heart of the Mendelow's matrix (1991), where the influence of an interested person is estimated as the product of their power index to the interest index. The scale of influence is worked out by each organization independently. This model, like the Mendelow model (1991), orders stakeholders by their influence but looks as follows in Figure 6.

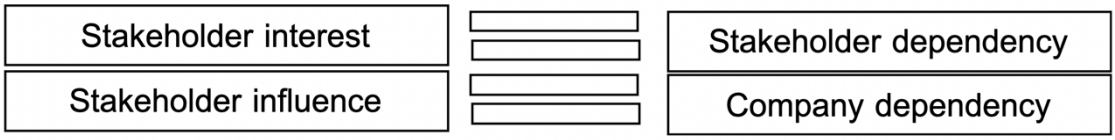


Figure 6. Incentive-influence and dependency relation (adapted from Mendelow's matrix, 1991, Thompson 2001)

The directions for interaction with stakeholders identified depending on the rating (Thompson 2001). The stakeholders of an open-source project have an interest in it as well as able to influence the result. Stakeholders' participation in the community is the motivation for developers, and their contribution to development, testing, and other areas of code improvement is invaluable. Therefore, this direction of evaluation is crucial for this research.

3. The balance model of resource relations (Rowley 1997). This model is based on the assumption that all stakeholders of any organization start relationships with each other in order to exchange resources valuable for them (Miles 2012, 64). As a result of these relationships, the network is created. The network representation of stakeholder relationships is shown in Figure 7.

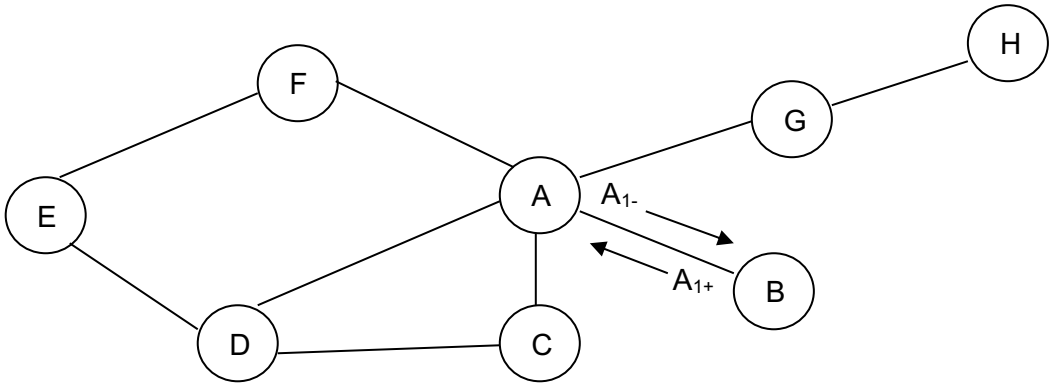


Figure 7. Network representation of stakeholder relationships (adapted from Rowley 1997)

It is necessary to explain the model where

- "A" is an organization,
- "A1+" is the resource that "A" gets from "B."
- "A1-" is the resource that "A" provides with "B."

Keeping the connection between network elements means sharing resources. There are three types of resource exchanges, regardless of their type:

- Asymmetrical in favor of the organization. It regards as the most advantageous option for the organization;
- Asymmetrical to the detriment of the organization. It is an unacceptable option, as it is not beneficial to the organization because organisation loses more than it receives;
- Equivalent. This option shows equal benefits for the parties (Miles 2012, 64).

The presence of any asymmetrical exchange can account for conflicts between the organization and stakeholders, so the system should strive for an equivalent exchange balance (Rowley 1997). This theory is not ideal to use for the evaluation of stakeholders of an open-source project, as the exchange of resources in such projects is often unequal.

4. The network model (Rowley 1997). In this model, as in the balance model, the relationship between the organization and stakeholders is introduced as a network. Within this theory, the position of an element in a network is described with the help of density parameters (a characteristic of the entire network, determining how much it is connected) and the elements' centrality (a characteristic of a network element, reflecting its position to others). Three indicators measure the centrality (Rowley 1997):

- Rank (the number of links that connect one element to another);
- Availability (the minimum number of connections from the element to all others);
- Intermediacy (the element able to be an intermediary);

- The frequency which enables the element to become an intermediary between any two elements.

Network density affects the potential possibility of stakeholders' manipulation. The higher the density, the more limited the resource flows, and the greater the influence of the element capable of controlling them and vice versa. Network density is directly related to centrality, because the stronger the stakeholder is connected to others and the more opportunities it has to mediate, the more likely it is that resource flows will pass through it, which in turn allows the stakeholder to have compensation for transit and more information. The organization strategy should be to establish as many connections as possible with stakeholders (network elements) when using a network model. Also, it should avoid a large number of intermediaries between the organization and direct stakeholders (Rowley 1997). This model may find its application in open-source projects, but its application is limited.

Analyzing the models given in this work, it is obvious that the Mendelow's matrix is the most efficient because the system proposed by the author is intuitive and applicable to the features of the open-source project. It has also found its application in the activities of leading stakeholder accounting organizations. Additionally, within the framework of this model, it is possible to estimate more precisely the influence of stakeholders on the OS project development.

In summary, one of the key challenges in organizing communication within the community of developers is a significant lack of prior research, the theoretical and practical knowledge about the portrait of the OS project developer audience. Earlier, researchers by Russian and American sociologists emphasized that a programmer (anyone, without any reference to a particular direction) is a reserved introvert-type professional (Anderson 2005) who avoids society and communication as such (Gray 1998). In more recent researches, this position has changed. Now the main characteristics of the representative of IT-area are:

- Mathematical thinking, a high degree of abstraction;
- Laconic nature (terseness of communication) of IT-area professionals;
- Difficulty to connect with multidisciplinary professionals;
- The gap in self-introduction skills;

- The specificity of terminology, used by programmers, "language barrier", prevail of English language in the professional slang, and terms (Hu, Zhao, Cheng, 2012);

- The young age of IT professionals, and, consequently, unwillingness to subordinate i.e. rejection of hierarchical structure, self-focusing, etc.

- Rejection of the for the authoritarian system of governance, preference for democracy and modern project management systems (Simonite 2018);

- Neglect of the dress code, striving for freedom of views, rights (Dixon 2018);

- Turning introversion into extraversion, i.e. the emergence of the need to communicate with people with similar interests) (Chydenius, Gaisch 2016).

Since a programmer spends most of his working time coding, and her only companions are "methods, functions, objects, modules, packages, etc.", so that the level of social interaction among programmers is usually lower than in other areas of intellectual work. Consequently, a low level of social interaction with people, according to studies of the past ten years, leads to professional burnout of developers, due to the change of introversion to extraversion (Brandford, 2018). Occupational burnout is defined as the total or partial loss of efficiency in the workplace due to increased emotional and physical exhaustion. It is manifested by growing indifference to their responsibilities and what happens at work, growing negativism towards both clients and colleagues, a sense of their professional failure, dissatisfaction with work. The recent research done by the biggest IT resource in Russia habr.com (2500 experts surveyed) showed that more than 50% of IT-professionals have experienced professional burnout, half of them have gone through this experience two or more times. For the employer, this kind of burnout of employees has quite serious consequences: up to 20% of employees are in this condition regularly, only 25% of burnouts remain at their previous place of work. This means that a large number of employees may work extremely inefficiently and interfere with others. Furthermore, there is a constant need to invest in the recruitment and adaptation of new employees to replace burnt-out ones. Employers should learn how to manage a burnout process to save the financial and human resources for the organisation (Karakulov 2019).

This trend has not bypassed open-source projects. According to Pieter Hintjens, professional burnout is also inherited by open-source projects. This phenomenon manifests itself in a deep disgust with the project so that the project is abandoned, contacts with participants broke up (Hintjens 2018). This type of burnout is like a

reckoning. Overcoming the burnout of open-source project developers can be solved in the following way:

- Reducing the level of responsibility and redistribution, redelegation of responsibility;
- Drafting and following a business plan;
- Attracting additional assistance with the project (Chasinga 2020, 24).

The community might become a solution to help cope with the burnout issue. Assessing the development of communities of OS projects, it is clear that the involvement of stakeholders in the development of the project contributes to productive and efficient work on the software (Hu, Zhao, & Cheng, 2012). The implementation of a mix of theories (life-cycle theory, stakeholder theory, and professional transformation theory) enables the establishment of an open-source project community that is able to contribute to the reduction of professional burnout (distribution of responsibilities) and prevent the stage of decline in the life cycle (keeping the stakeholders motivated and engaged by introducing online and offline forms of interaction). Stakeholder theory determines the groups the community should be designed for in order to identify and engage the priority groups in empirical research based on the theoretical analysis.

3 The empirical research

This chapter describes the research methodology as well as key data collection methods. This chapter also provides the framework of the research and tools used to get the valid results.

3.1 Methodology

First of all, it should be noted that the philosophy of the planned research is pragmatism, the form of research where the practical result and research issue are the basis for the conclusions. It is recognized within the framework of this philosophy that there is a number of possibilities to understand the overall picture and conduct the research based on the problem.

The goal of this research project is to study certain issues and organize it in a new conceptual framework of the way to develop communication to increase the level of interaction with the stakeholders of a project, the number of contacts, and their engagement rate into an open-source project. These effects as a whole should become the basis and growth driver for the developers' community.

It should be pointed out that the ways to engage stakeholders in a project are numerous; they depend both on project features as well as on personal and professional interests. Hence, pragmatism seems to be the best choice because the obtained result can be significantly different from the author's personal perspective or the settings of the existing core of an open-source project.

As mentioned above, the goal of this study is to develop a model on a conceptual basis and therefore the deductive approach was chosen to solve the problem. The deductive research is based on a combination of existing theoretical hypotheses; the inductive research generates a new theory. Through the use of the deductive approach, author can formulate a hypothesis based on the existing theory.

The key methodology of the research is constructive approach, aimed to improve existing systems, transit from existing knowledge to a new model and practical developments to improve knowledge in the chosen direction. The use of the constructive approach can be noted in the structure of this research, as one can see in Figure 8.

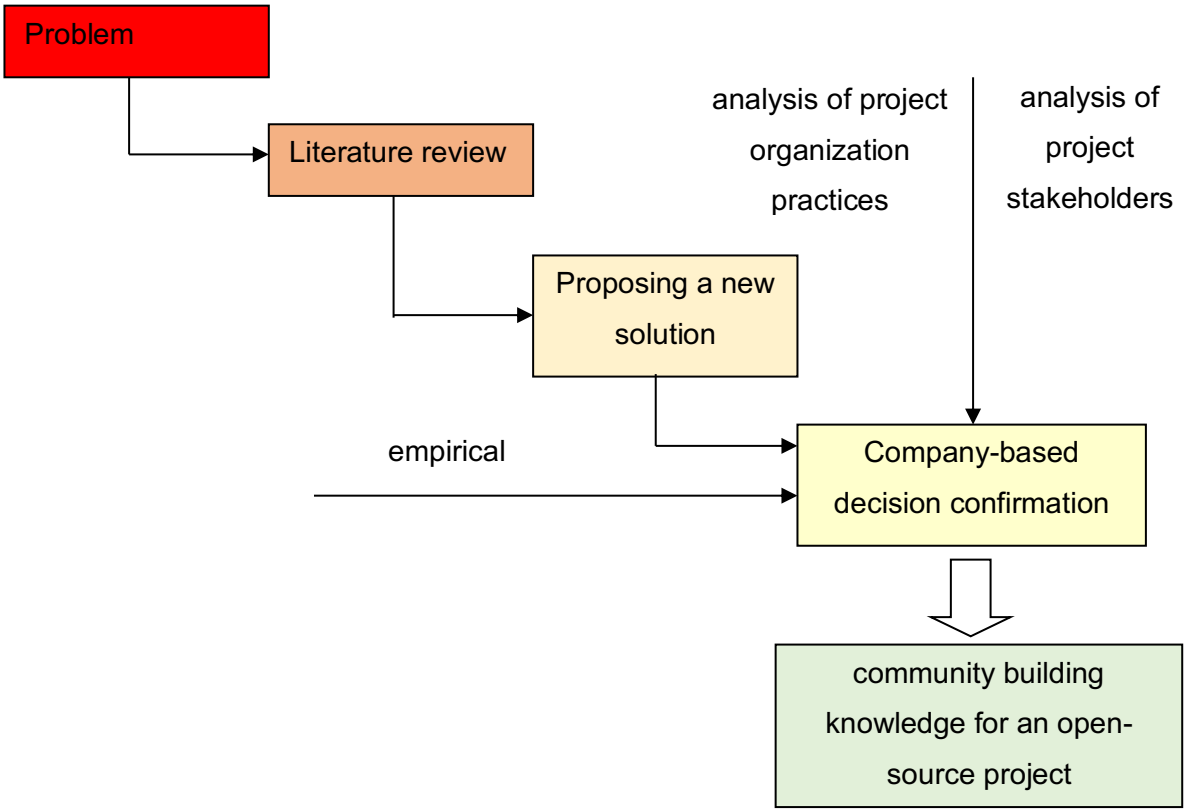


Figure 8. Research structure

From the existing methodologies, it seems that constructive research is the most appropriate model, as it aims at developing novel solutions based on existing knowledge. This approach is aimed at producing an innovative construction tacking a particular practical issue (Lukka 2003).

This type of research involves “design thinking which makes projection into the future envisaged solution” (Magnani, Carnielli, Pizzi, 2010). The solution can be presented in a form of guidelines, models, plans, artifacts, diagrams, algorithms, etc.

Constructive research is a problem-solving approach that is used to improve the existing system, or performance, fill in knowledge gaps. There is a conflict between constructive and scientific problem-solving methods. In scientific research decision-maker usually gives the aims to the researcher who finds the solution using scientific methods whereas constructive approach implies that researcher engaged with designing constructs as a way of producing solutions never existed before, which can be demonstrated only in a new reality. Often constructive research encourages co-production between the industry and the researcher (Oyegoke 2011).

Quantitative and qualitative research methods are traditionally used to solve the research problem. This research presents a synthesis of these two methods. This is

explained by the following: any open-source project has the core, its closest environment, and the external environment, that can be seen in Figure 9.

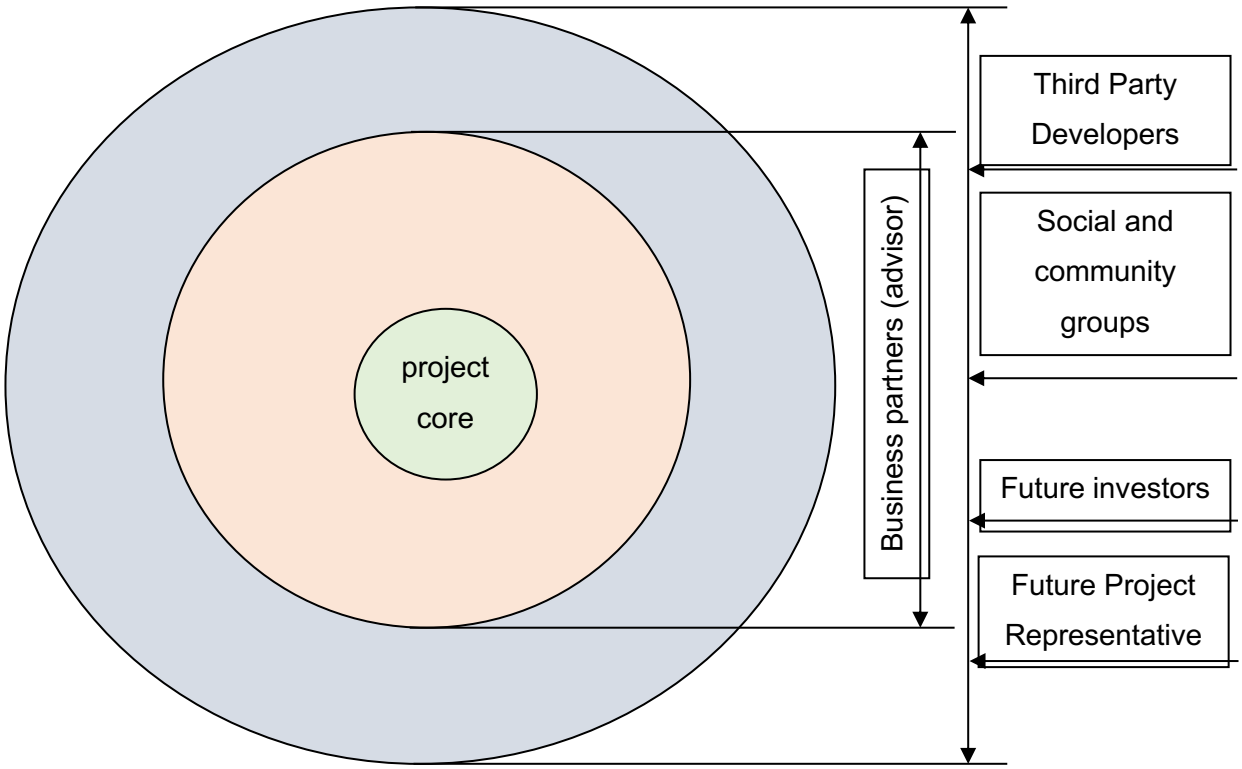


Figure 9. Case study project structure

Thus, analyzing the data demonstrated in the Figure 9, open-source projects' core usually is rather small. The closest external environment (microenvironment) is also numerically insignificant and static (engaged advisors). The macroenvironment of an open-source project is dynamic and global.

To get valid results of the research, it is optimal to engage both the developers (the core) of the analyzed project and the cores of similar projects.

The qualitative research method helps extract the data of the choices made by core developers concerning the forms of communication. Taking into consideration that core groups are the most interested in communication with the stakeholders of a project as well as numerically insignificant, these results should become the basis of the research.

The quantitative method was chosen to study the global community as it provides the answer to the research problem given by the various representatives of the external environment.

The synthesis of qualitative and quantitative methods allows obtaining valid results regarding the form of communication for the community and draw the conclusions.

3.2 Research design

This subchapter illustrates the logic, process and structure of the empirical research.

The research design is determined by its purpose and tasks, as well as the forms selected for it, of which there are three:

- Focus group study;
- Interviewing;
- Online survey.

The research objectives were:

- To identify the participation of the respondents in open-source software development projects;
- To estimate the time of participation of the respondents in such projects;
- To explore the interaction of the respondents with the project;
- To analyze the role of the respondents in those projects;
- To assess the importance of creating a community as a platform for interaction, according to the respondents;
- To assess the importance of communication in the community;
- To identify the preferred form of communication within the community;
- To study the recommendations for introducing a certain form of communication into the community.

Therefore, based on these tasks, the empirical indicators of the research are:

- The participation of the respondents in open source-software development projects;
- The time of the respondents' participation in such projects;
- The interaction of the respondents with the project;
- The role of the respondents in such projects;
- The importance of creating a community as a platform for interaction, according to the respondents;

- The importance of communication in the community;
- The form of communication within the community;
- Recommendations on the introduction of a certain form of communication into the community.

In general terms, the stages of the empirical research include:

1. Defining the methodological framework for the study: research goals, objectives, hypotheses;
2. Selecting the methods of information collection that adequately meet the goal set and tasks to be solved;
3. Developing research tools (questionnaires for the research);
4. Making the survey publicly available, providing the participants with a link to it;
5. Conducting a focus group study;
6. Conducting a series of interviews;
7. Processing of the obtained empirical material;
8. Describing the results obtained;
9. Analyzing the data obtained;
10. Developing recommendations in accordance with the results of the study;
11. Drawing the conclusions.

The conducted empirical research on the side of the researcher is based on a theoretical analysis of the literature on the research topic, the study of the concept of “community”, as well as on the analysis of communication features in the IT area, the characteristics of open-source projects.

To conduct an online survey, 7 questions were proposed, including:

- Six closed questions (answer options provided);
- One open question (with the option for the respondent to provide a response).

The proposed design for the online survey questionnaire (quantitative form) is presented in Figure 10.

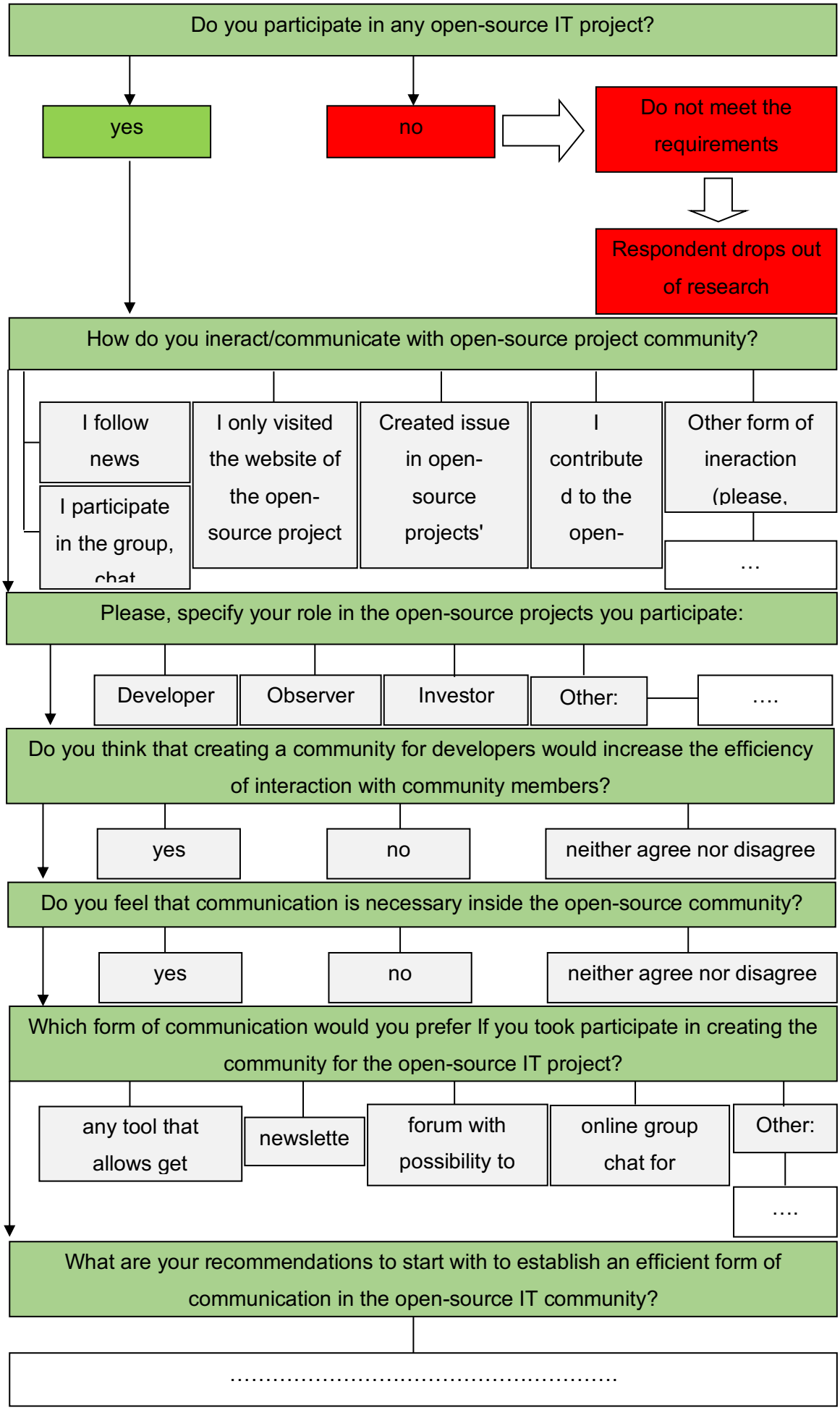

Do you participate in any open-source IT project?
yes
no
Do not meet the requirements
Respondent drops out of research
How do you ineract/communicate with open-source project community?
I follow news
I participate in the group, chat
I only visited the website of the open-source project
Created issue in open-source projects'
I contributed to the open-
Other form of ineraction (please,
…
Please, specify your role in the open-source projects you participate:
Developer
Observer
Investor
Other:
….
Do you think that creating a community for developers would increase the efficiency of interaction with community members?
yes
no
neither agree nor disagree
Do you feel that communication is necessary inside the open-source community?
yes
no
neither agree nor disagree
Which form of communication would you prefer If you took participate in creating the community for the open-source IT project?
any tool that allows get
newslette
forum with possibility to
online group chat for
Other:
….
What are your recommendations to start with to establish an efficient form of communication in the open-source IT community?
……

Figure 10. Design of the online survey questionnaire (quantitative research form)

As can be seen from the figure, the first question is aimed at screening out respondents who have not previously participated in projects, do not have the necessary experience in such projects. In general, this study, as noted earlier, is directed to the external environment, or to external stakeholders of the company.

- Social and community groups.

The results obtained during the study will be supported by the findings of the focus group study, as well as additional interviews with the “core” of the projects.

As noted earlier, the focus group study sample is represented by 4 groups of respondents. These groups of respondents include:

- 4 employees of the Fluence project team (Focus Group 1);
- 4 employees of the Cyberdevelopment project team (Focus Group 2);
- 4 employees of the project team Smart Technologies (Focus Group 3);
- 4 employees of the Infocompas project team (Focus Group 4).

The proposed focus group study scenario is illustrated in Figure 11.

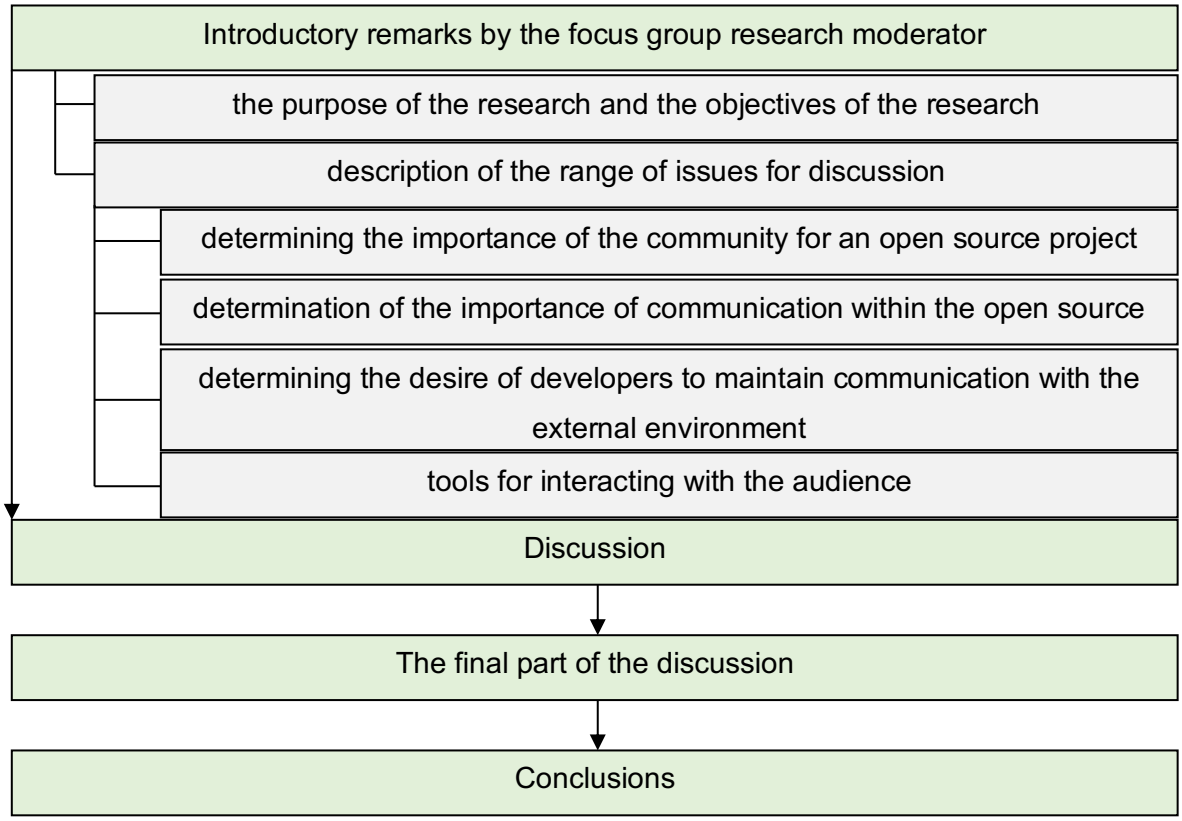


Figure 11. Design for the focus group research (qualitative research form)

During the conversations and discussions, valuable knowledge will be extracted, the best practices for organizing the communication between the core of a project and the environment will be identified. It should be noted that as a negative effect of the research, it can be revealed that open-source projects developers do not need a community, because the core is self-sufficient and closed. The four groups were invited in order for the result (even if negative) to be unbiased.

Additional interviews are applicable in order to reveal the developers' personal preferences and wishes. It is possible for the researcher in the set of a private conversation to get new results, the respondent may be psychologically willing to reveal more. It is likely that during group conversations people might not be able to make suggestions that are of value to the research. The questions planned for the interview are shown in Figure 12.

Interview Questions

1. What is the status of your project?

2. How long have you been working on your project?

3. In your opinion, is it necessary to create a community for communication between participants in the development of open source software? I would like to hear your thoughts on this.

4. Is communication a special platform or meetings with participants and stakeholders within your project important for you? Why?

5. What is the most optimal interaction (communication) tool for an open-source project for you?

6. What are the main recommendations for the implementation of the optimal form of

Figure 12. Interview design (quantitative research)

Sixteen respondents will participate in the interview which are representatives of the studied companies, so-called "cores" of an open-source projects.

It should be noted that all the answers obtained as a result of the empirical research are based on the knowledge and personal experience of the respondents.

The next chapter of this work will highlight in detail the results obtained during the study, as well as their synthesis, convergence and interaction.

The methodology discussed in the given chapter was used for the research, the proposed methods were applied, and accordingly, valid results were obtained.

3.3 Data Collection

This subchapter highlights the qualitative and quantitative data collection methods, tools and structure.

Data collection for the research is represented in two forms:

- Primary data collection;
- Secondary data collection.

Two methods of research are used to collect primary data:

1. Focus group research;
2. Online survey on a special platform.

1. Focus group research as a method of obtaining primary data for qualitative research.

In order to obtain valid data, representatives of four companies involved in open-source software development were engaged in the study. The participants for the focus group study are chosen from the following organizations (Table 1):

- Fluence employees, the given company served as the research base;
- Cyberdevelopment employees;
- Smart Technologies employees;
- Infocompas employees.

Table 1. General description of the focus groups participating in the study

Group	Focus group 1: Fluence	Focus group 2: Cyberdevelopment	Focus group 3: Smart Technologies	Focus group 4: Infocompas

Number of participants		4 pers.	4 pers.	4 pers.	4 pers.
Form of participation in the project	Developer	4	2	3	2
	Project Manager	0	1	0	1
	Release Engineer (DevOps)	0	0	1	1
	Product Owner	0	1	0	0

Thus, each of the focus groups was represented by the same sample of participants (4 people) related to the product. The specific structure of the interview participants is shown in Figure 13.

The structure of the participants of the focus group study is shown in Figure 13.

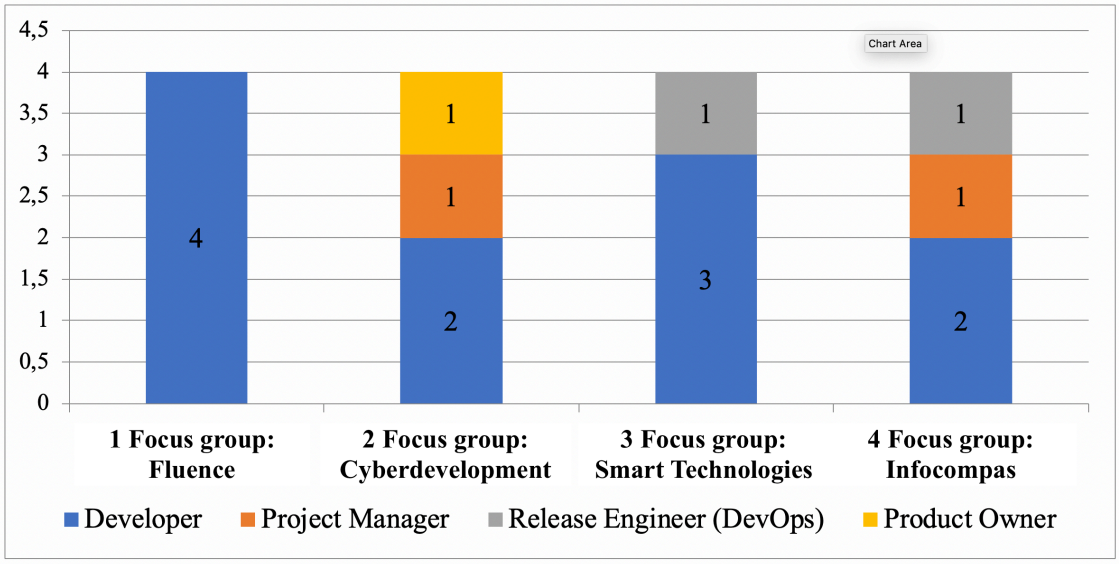


Figure 13. The structure of the focus group study participants

In order to be valid and reliable the sample for the research is represented by variety of roles in the open-source project.

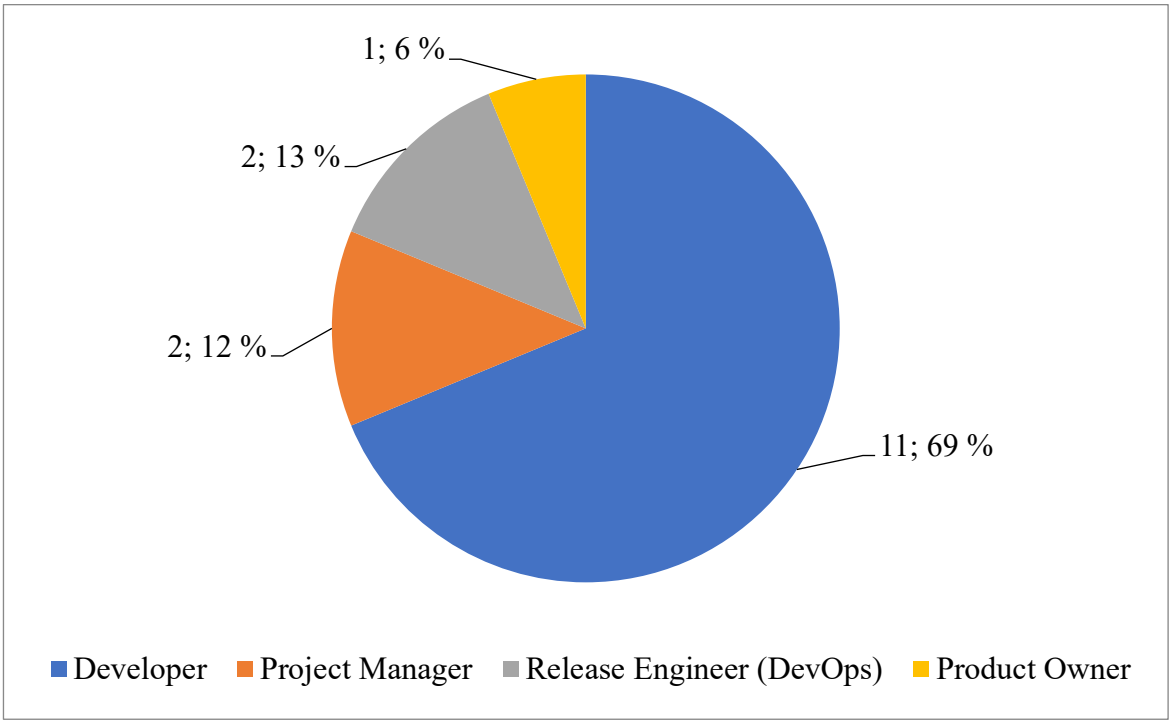


Figure 14. The roles represented in focus groups

Assessing the presented roles structure, we see that most of the participants in the focus group study belong to the “Developer” category: 69% of the participants. The second largest group includes the “Project Manager” and “Release Engineer” categories.

It should be pointed out that all participants belong to the “core of the project” category, i.e. the open-source software project team (or 100% of the participants are the target group of the study).

The focus group research method was applied to collect data from this group. The method presented in the form of a conversation followed by an additional interview with each of the participants.

The main characteristics of the data collection stage of the qualitative research are shown in Table 2.

Table 2. The main characteristics of the primary data collection stage of the qualitative research

Research Base	Date	Time	Stage Characteristics
Focus group 1: Fluence	April 2, 2020	12.00 – 14.00	General conversation with the study participants organized in the form of a video chat

		15.00 – 15.30	Interview with respondent 1
		15.30 – 16.00	Interview with respondent 2
		16.00 – 16.30	Interview with respondent 3
		16.30 – 17.00	Interview with respondent 4
		17.00 – 19.00	Processing of the results
Focus group 2: Cyberdevelopment	April 3, 2020	12.00 – 14.00	General conversation with the study participants organized in the form of a video chat
		15.00 – 15.30	Interview with respondent 1
		15.30 – 16.00	Interview with respondent 2
		16.00 – 16.30	Interview with respondent 3
		16.30 – 17.00	Interview with respondent 4
		17.00 – 19.00	Processing of the results
Focus group 3: Smart Technologies	April 4, 2020	12.00 – 14.00	General conversation with the study participants organized in the form of a video chat
		15.00 – 15.30	Interview with respondent 1
		15.30 – 16.00	Interview with respondent 2
		16.00 – 16.30	Interview with respondent 3
		16.30 – 17.00	Interview with respondent 4
		17.00 – 19.00	Processing of the results
Focus group 4: Infocompas	April 5, 2020	12.00 – 14.00	General conversation with the study participants organized in the form of a video chat
		15.00 – 15.30	Interview with respondent 1
		15.30 – 16.00	Interview with respondent 2
		16.00 – 16.30	Interview with respondent 3
		16.30 – 17.00	Interview with respondent 4
		17.00 – 19.00	Processing of the results
Preparation of transcripts of the interviews with each of the participants	April 9, 2020 – April 13, 2020	9.00 – 12.00, 17.00 – 20.00	Processing the obtained empirical material, bringing it into a readable form in order to be added to the paper
Processing of the received material	April 16, 2020 – April 18, 2020	9.00 – 17.00	Examining the context of the empirical material collected

Formulation of conclusions	April 19, 2020 – April 20, 2020	9.00 – 17.00	Drawing conclusions on the obtained empirical material of the qualitative research

Graphically, this data collection process for the qualitative research is shown in Figure 15.

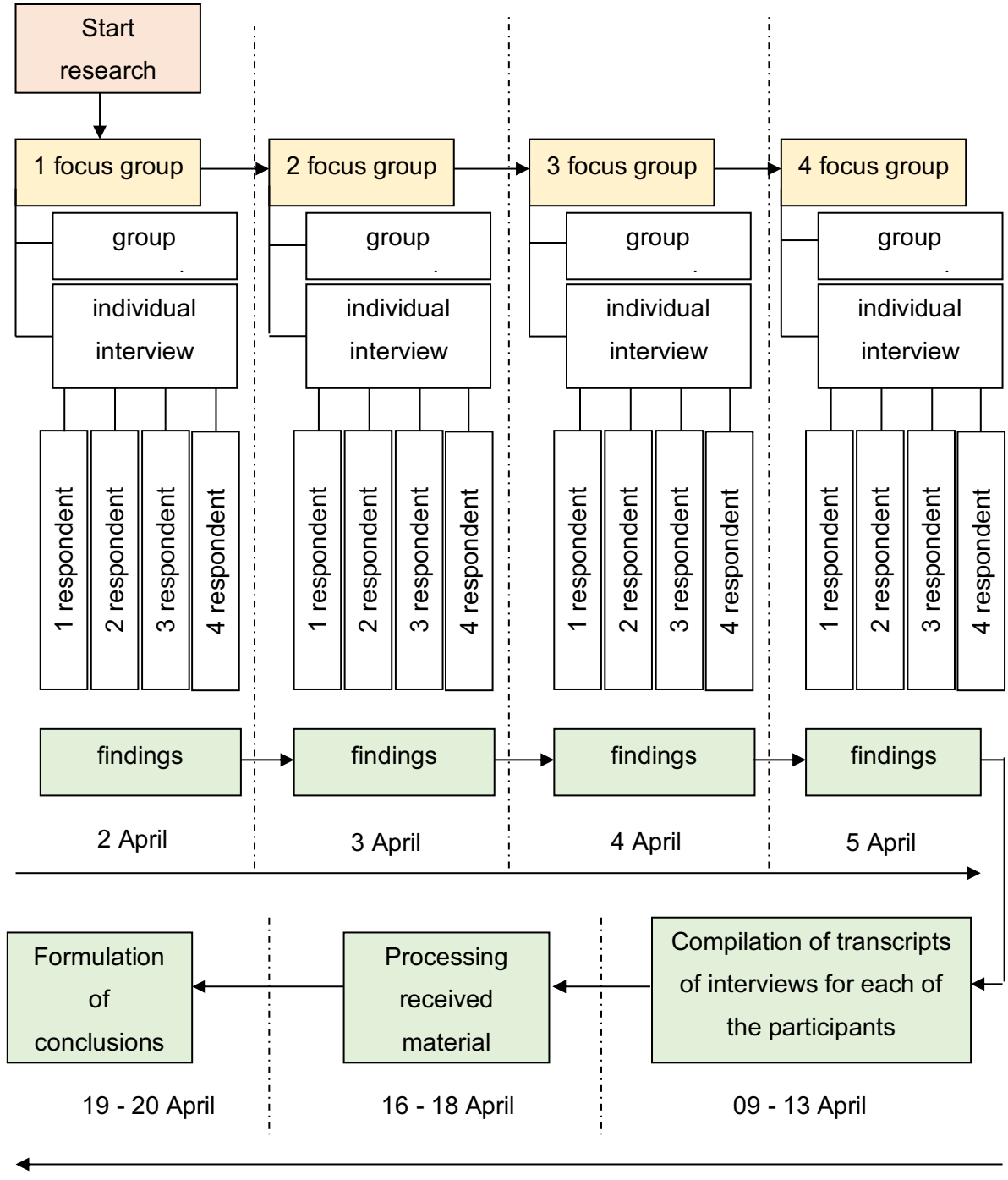


Figure 15. Primary data collection for the qualitative stage of the study

Thus, the qualitative research is represented by both a general group study and a series of interviews. In total, the study involved 4 groups of 4 people (or 16 respondents). The study took place from April 2, 2020 to April 20, 2020 (19 days).

It is important to recognize the nature of this type of companies. Software development organizations are normally geographically distributed. Hence, the data collection for the empirical study was carried out online, using a special video conferencing application, Zoom (Figure 16). The given service allows to make free video calls. The program has a nice and easy user interface with possibility of chat for providing the useful links as well as recording the interview (Martinez, March 2020).

Figure 16. The software selected for the focus group study (adapted from https://zapier.com/blog/effective-remote-meetings/)

The service works directly from a browser, does not require download or registration, however app is convenient to use as well. The application automatically generates a link to the channel for video communication; this link was provided to the participants one day before the data collection.

It was not by accident that we have chosen lunch time for the group study: according to the personal experience of the author of this paper, precisely in the period from 12:00 to 14:00 we observed low levels of the application users activity, which made it possible to effectively hold a video conference with the participants without any technical problems. For individual conversations, the time does not play a significant role: the signal is stable;

problems are encountered only during group conferences, and only in the morning (before 11:00) and evening (after 15:00) time.

The primary data collection was carried out in the manner presented above, in order to conduct qualitative research in the format of a focus group.

2. Online survey as a method of obtaining primary data for quantitative research.

As was proved earlier, the external environment of open source projects is numerous and constitutes a global society. To study the characteristics of the planned community, an online survey was created on a special platform for conducting such studies, Survio. This tool allows to create online surveys and post a link to them on specific platforms (or share it privately). This application is convenient, as it provides the possibility to create polls based on ready-made templates, keep replies, analyze statistics (Figure 17).

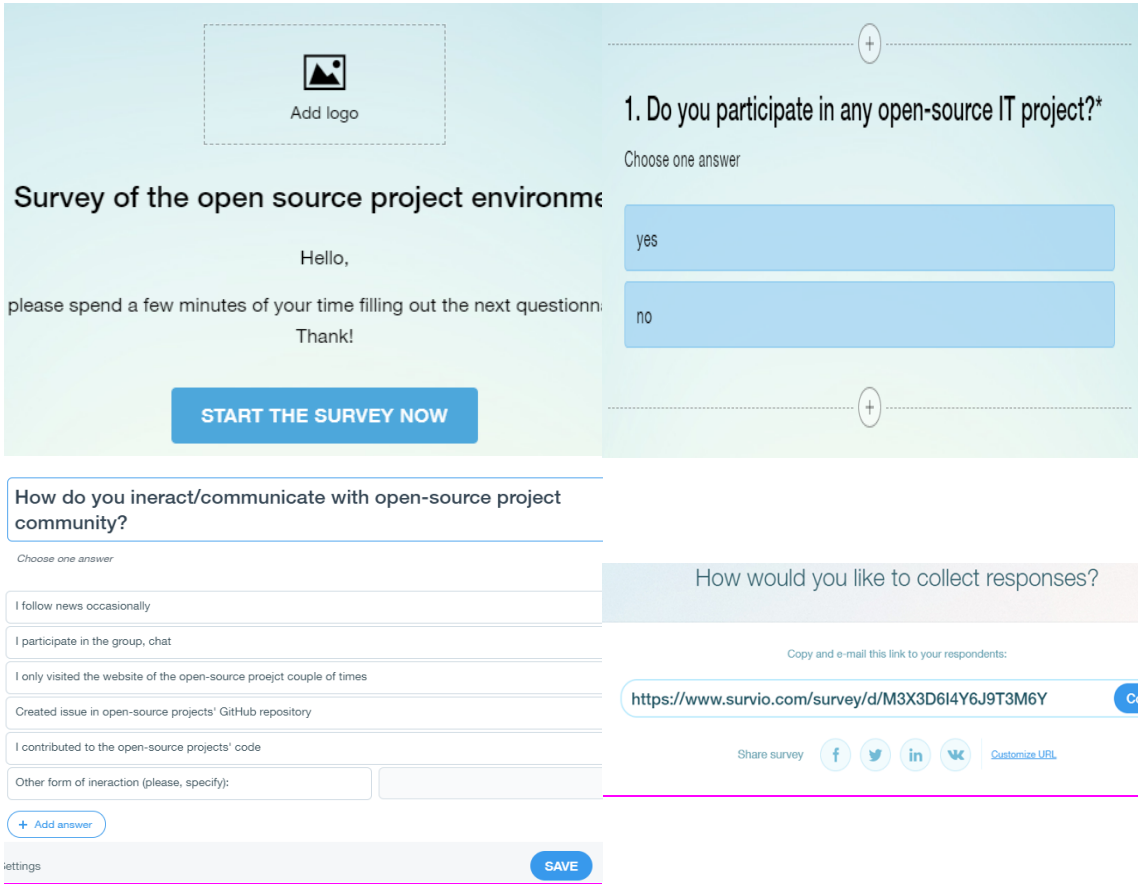


Figure 17. The use of Survio as an online survey platform, adapted from survio.com

On the part of the user (respondent), the survey UX is convenient: the service interface is adapted for different operating systems and is optimal for accessing it from a mobile device (Figure 18).

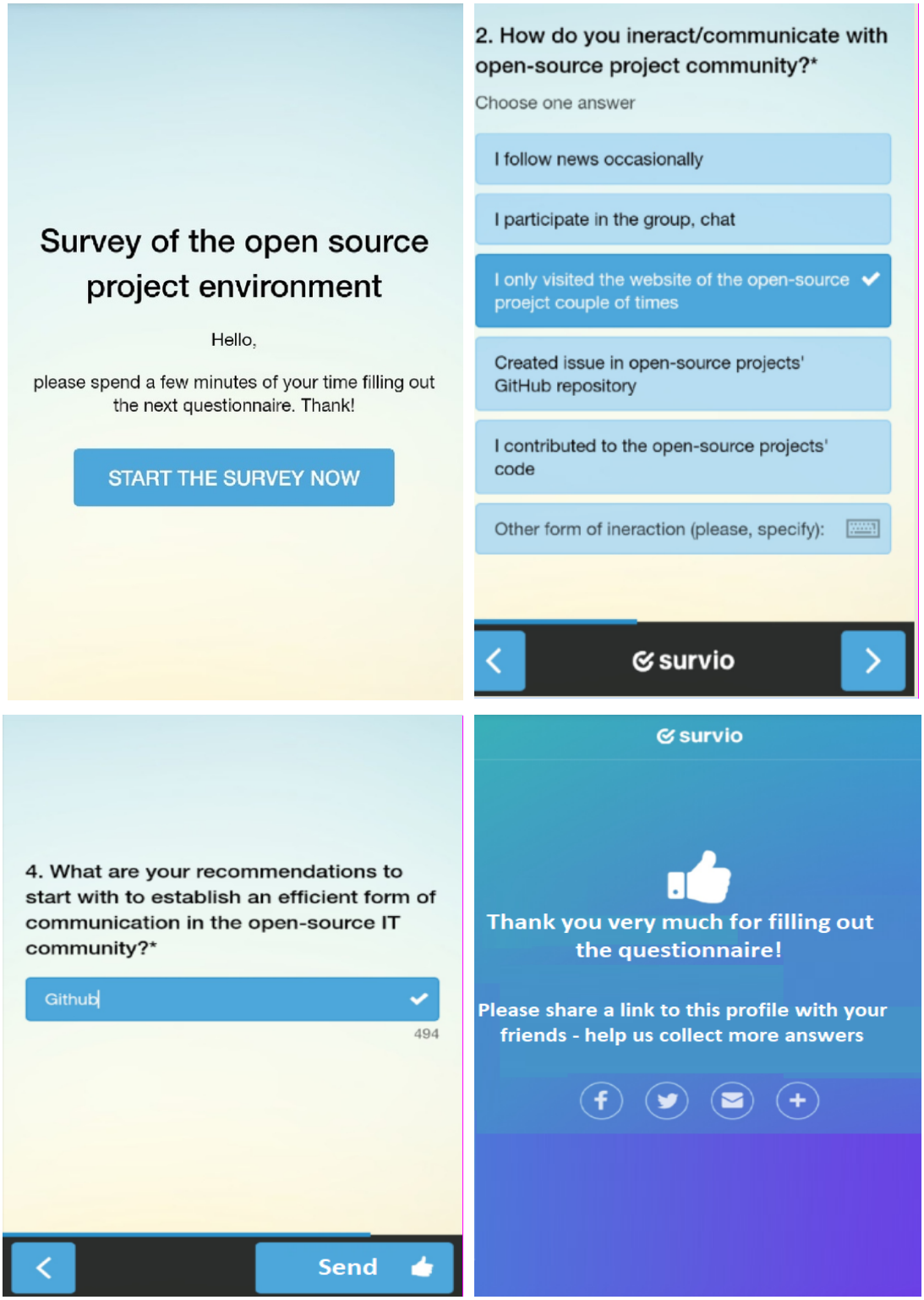


Figure 18. Quantitative survey software interface from the user (respondent) side, adapted from survio.com

To achieve the objective of the research, a survey was created on the Survio site that included both open and closed responses. The link to this survey was posted in various open sources for data collection:

- In developer forums;
- In social media;
- On profiled websites.

It should be noted that the survey also contained a “screening indicator”: respondents who were not suitable for the given research were screened out.

The quantitative research (the start of the survey) was carried out simultaneously with the qualitative research: on April 3, 2020. While the focus group study was under way, the respondents’ answers were collected. The survey was completed on April 20, 2020. The structure of the quantitative research is presented in Figure 19.

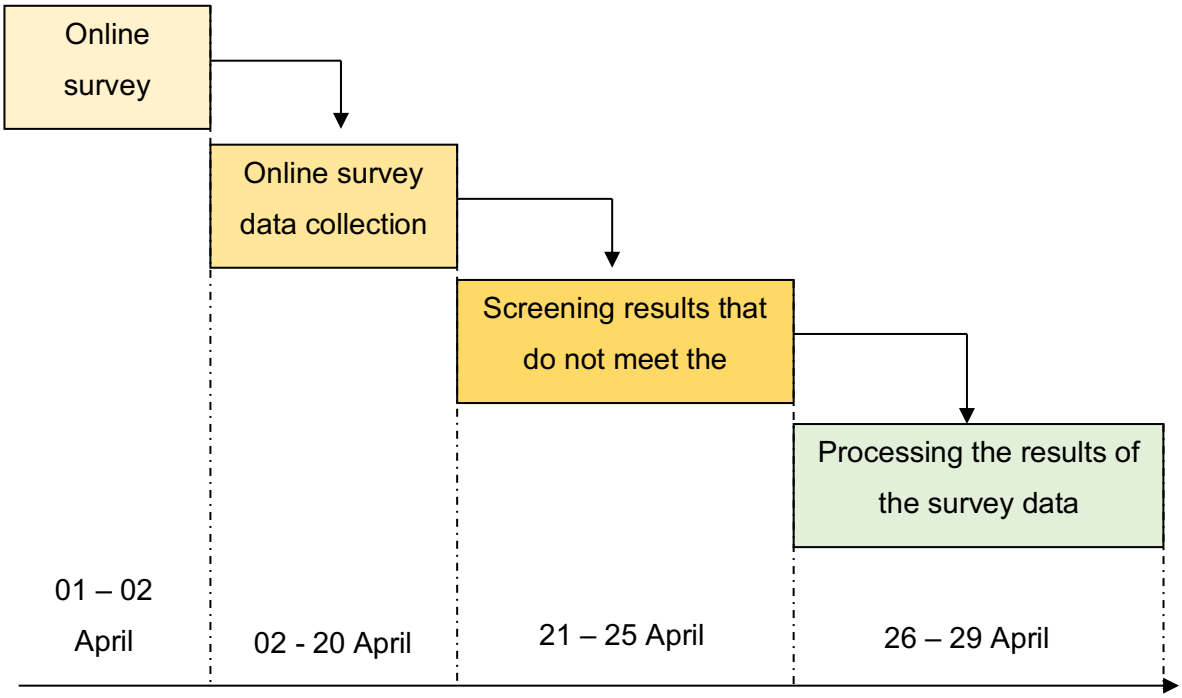


Figure 19. The structure of the quantitative study

Thus, the total time for conducting the quantitative study was 28 days, out of which the collection of the results by links (data accumulation) took 19 days.

The data acquisition structure (by days) is shown in Figure 20.

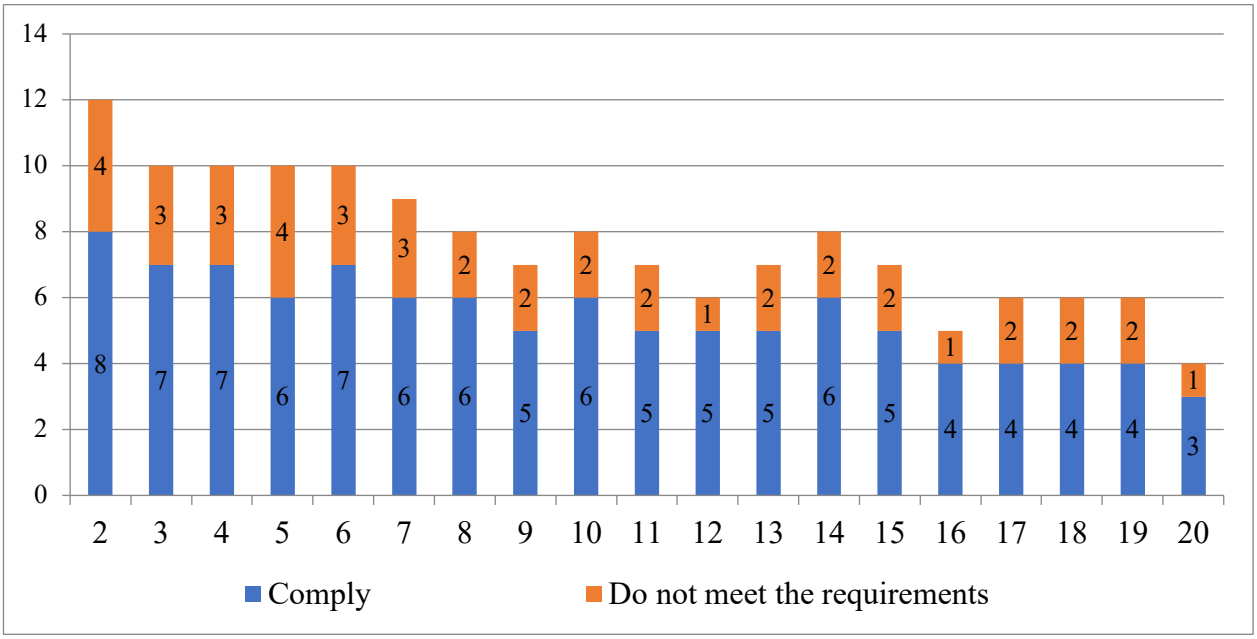


Figure 20. The structure of the responses received to the online survey, by day, pcs.

The resulting structure of the responses to the survey for the entire research period is shown in Figure 21.

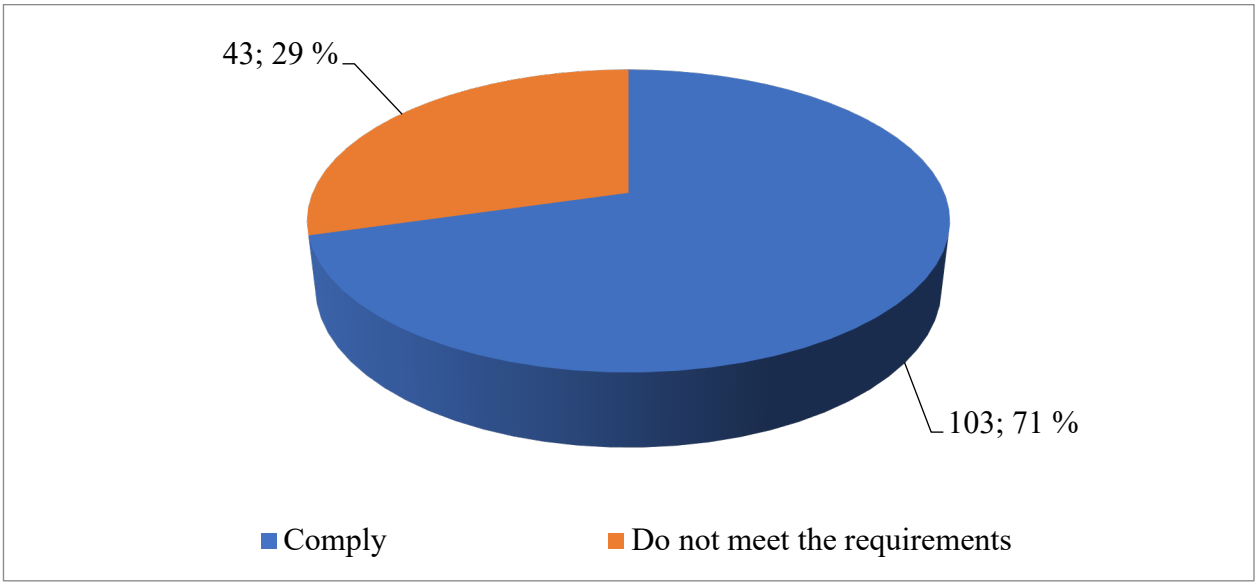


Figure 21. The resulting structure of the responses to the online survey for the entire research period

A total sample of 146 responses were collected for the survey (over 19 days of collecting responses), of which 103 responses fit the specific needs of the given research. 43 respondents had not participated in open-source software development projects (and in

global IT projects in general), so they were screened out for lack of the necessary experience.

Secondary data were also used to study the given topic. The secondary data included surveys conducted by the research base company, Fluence, which is an open source software development company. These studies do not have a direct focus on creating a community, but present several interesting points, for example:

- The availability of communication capabilities at existing platforms for collaborative software development;
- The number of core team members of various projects, their structure;
- The number of users of the project;
- Financing / funding details;
- Willingness to use a decentralized storage (or company product).

Fluence conducted a study (Fluence Labs, 2019) which included the above-mentioned parameters, which are in turn useful for drawing additional conclusions.

4 Findings

This chapter highlights the special aspects of the open-source startup project. The research of the special aspects supports finding the best ways of communication with community members as well as vital elements of the planned community.

Substantial part of this chapter is dedicated to defining the stakeholders of the OS startup project for the reason that stakeholders are the key element of the community in the case company. The study of the needs of the project as well as its environment is crucial for this work and establishment of the community which meet the requirements of any community member.

This chapter also discusses the particularities of the communication in the OS projects as well as the needs of the target audience. The applied research provides new knowledge of establishing a community of developers.

4.1 Definition of the target audience of open-source project and its needs

The characteristics of the target audience and the core of the OS projects are defined according to the following criteria taken from the research done in 2019 by the author of this thesis on board of the Fluence Labs team:

- The project foundation year;
- The size of the core team;
- The size of the project environment;
- The core programming languages of the OS project and the environment;
- The platforms used for work;
- The project financing / funding;
- The construction of communication between the core and the community (Fluence Labs, 2019).

The first area of market research is the foundation year of the OS projects. As part of the study carried out by Fluence Labs representatives, the following statistics on the OS projects foundation year were compiled, as shown in Figure 22.

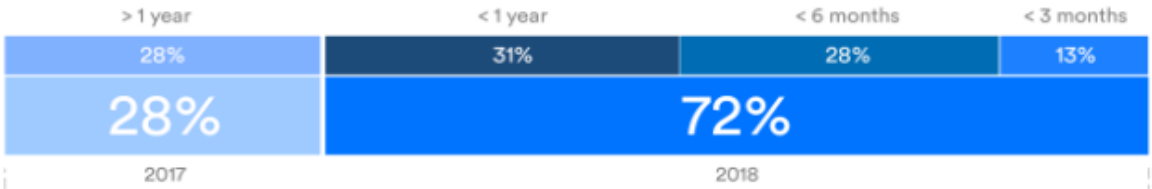


Figure 22. Open-source projects foundation years (adapted from Fluence Labs, 2019)

It should be noted that 160 open-source projects were researched. Most of the projects presented on the market exist for less than 1 year. Only 28% of all projects (or 45 projects) exist for longer than one year.

The size of the core team of open-source projects is shown in Figure 23.

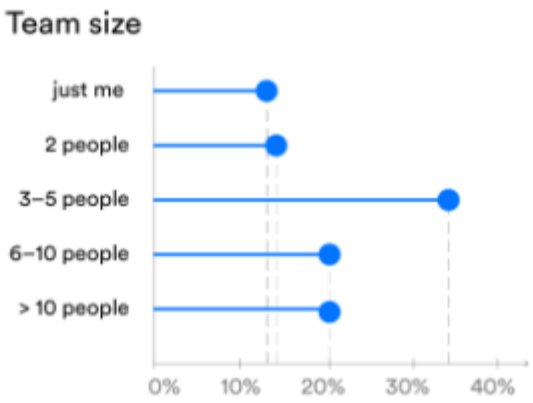


Figure 23. The size of the project core team (adapted from Fluence Labs, 2019)

Thus, only 12.5% of the studied projects were carried out by a solo developer. Most of the projects had a team size varying from two to five (47.5%), or more than five developers (20%).

The number of users of OS projects is shown in Figure 24.

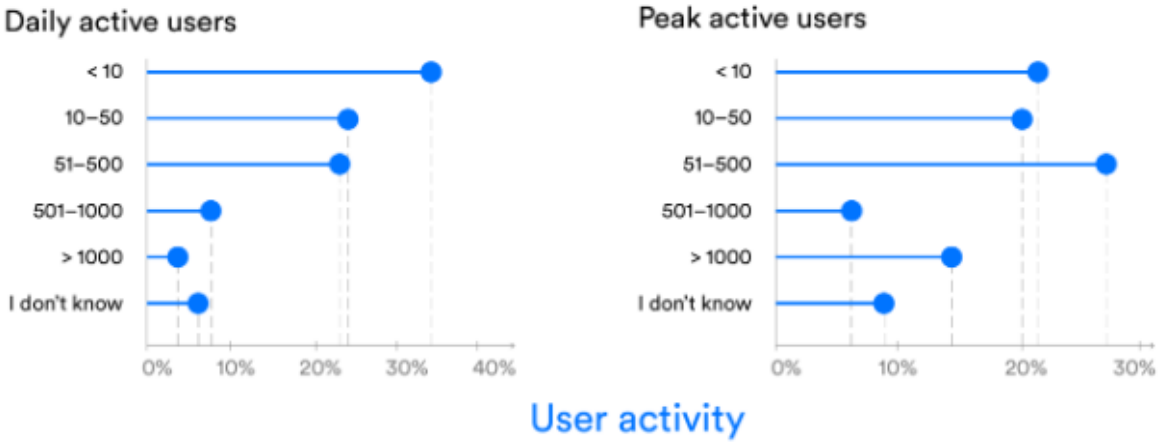


Figure 24. Number of the project users or size of the global environment (adapted from Fluence Labs, 2019)

Analyzing the data obtained, it was found that 58% of the projects have less than 50 daily active users, and 12% of the projects have more than 500 daily active users.

The programming languages used by the global environment, as well as the core of the project, were also researched. The results are shown in Figure 25.

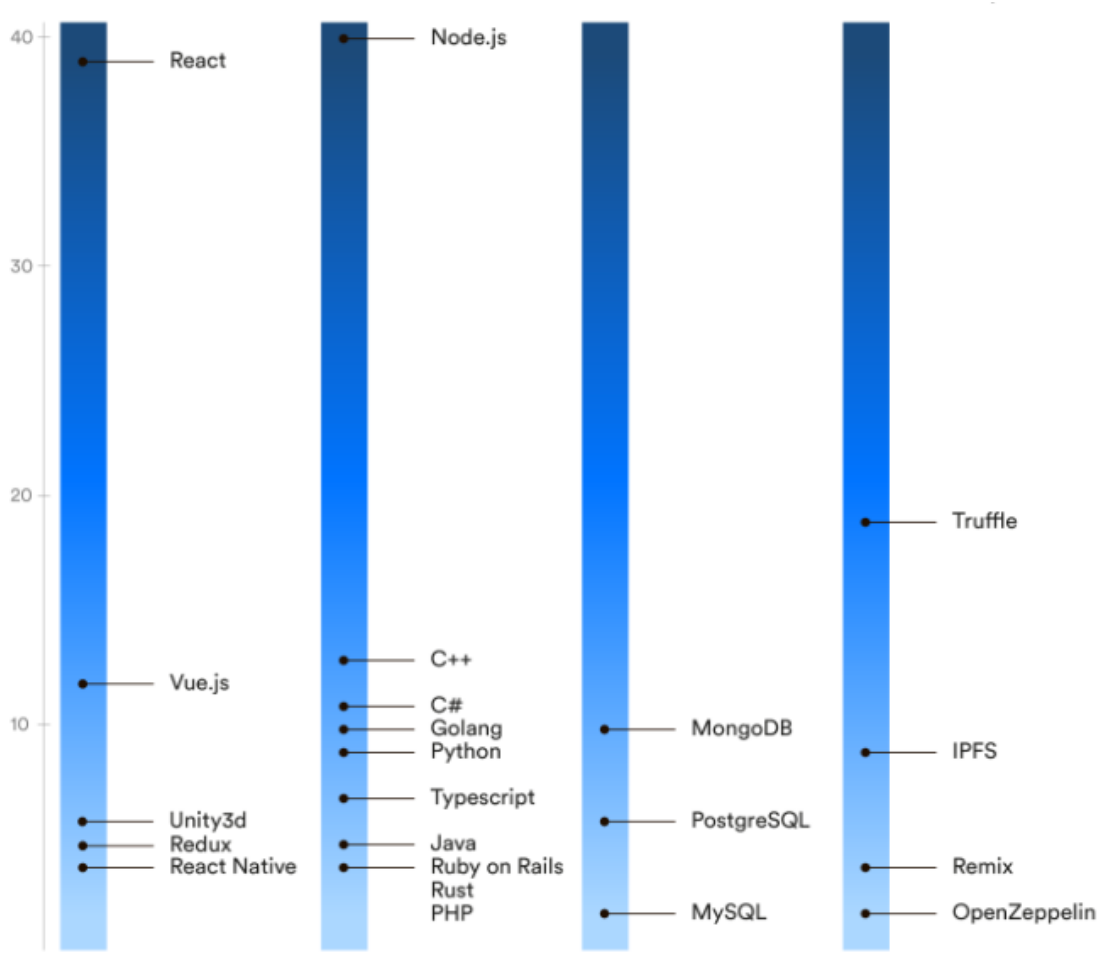


Figure 25. Programming languages used by the project core and global environment (adapted from Fluence Labs, 2019)

Thus, most representatives of the global environment use the React and Node.js languages, leaving behind other popular languages and frameworks.

A remarkable point is also the study of storage facilities used by both the developers and the environment. The results of this study are shown in Figure 26.

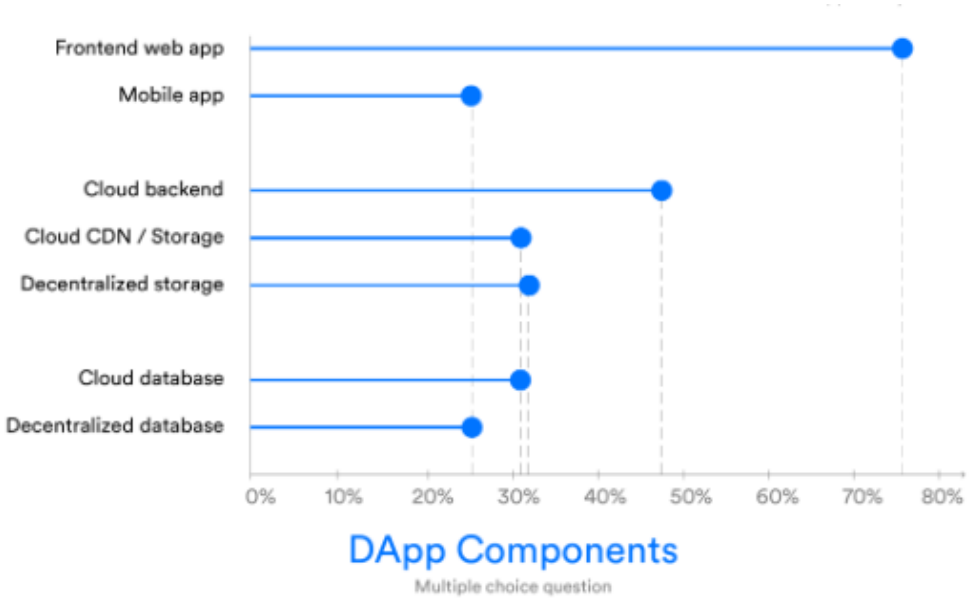


Figure 26. Types of storage used by the developers and the environment (adapted from Fluence Labs, 2019)

The data illustrates that the majority of the developers and their environment use traditional cloud storage for the code execution (48%). Decentralized storage is used by 32%. Centralized CDN storage accounts for 31%. 31% of the developers use centralized databases, while 25% of the respondents use decentralized databases. It should be pointed out that most of the respondents note that the use of centralized storage facilities is not convenient: they are unstable, incompatible with each other, unpredictable; frequent failures and disruptions are observed.

The statistics on the projects financing is shown in Figure 27.

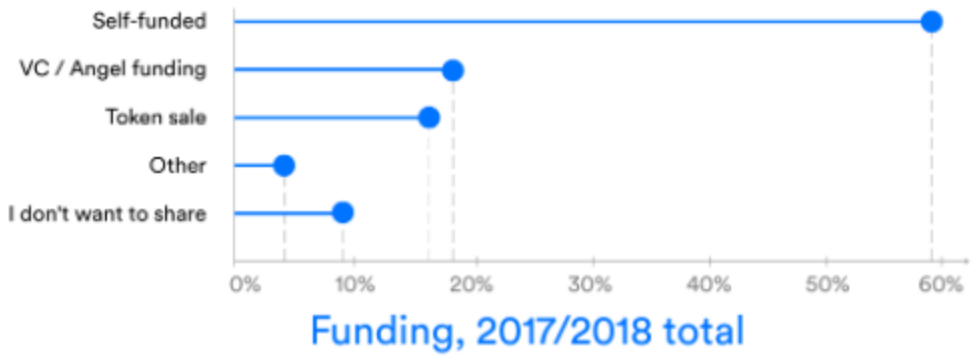


Figure 27. Statistics on the OS projects funding (adapted from Fluence Labs, 2019)

Mainly OS projects are self-funded (38%), or financed by token sales (31%). Projects that have investment from VC or angel funding accounted only for 24%.

Another point of the Fluence Labs (2019) research draws attention to the statistics of the communication between the developers and the external environment.

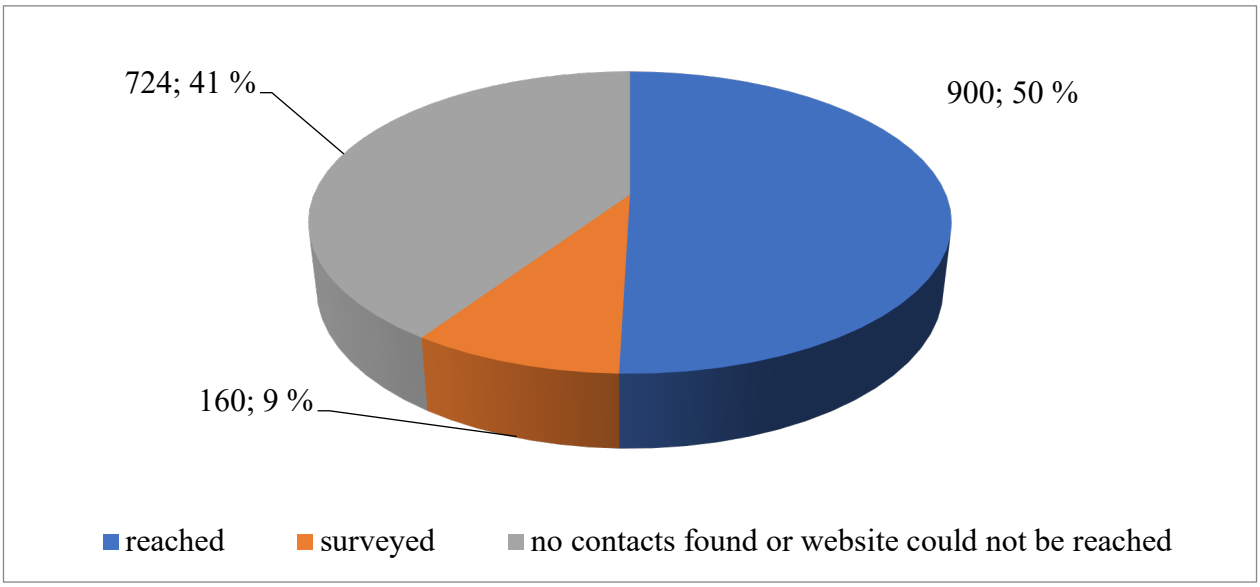


Figure 28. Communication between the project developers and the external environment

Based on an analysis of 1,724 projects, it was found that only 900 of them have e-mail, Telegram or Discord available for interaction between the project core and the stakeholders. This rate (50%) is significantly low.

Following on from the secondary data analysis, significant challenges revealed:

- There is a rising interest of the global environment in small OS projects (more than 500 users per day);
- The use of decentralized data storage facilities by the users, willingness to develop these due to the low trust to centralized storage;
- Lack of tools for building communication between the project core and the stakeholders (only 50% of the projects have feedback tools).

With the current trend, there is no community of developers to speak of: the conditions existing in most projects cannot become a basis for bringing people together, there is no communication between the two groups, and the needs of the target audience are not taken into account. It should also be noted that involving stakeholders in the interaction can become the basis not only for developing partnerships, creating a base of mutual assistance, cooperation and mutual support in the IT environment, but can also attract funding, which is strategically important for startup projects.

4.2 Defining the stakeholders of the open-source project

Currently Fluence Labs, which is implementing Fluence product, is represented by 6 employees, 4 of which are developers. Team includes:

- Evgeny Ponomarev (Co-founder);
- Dmitry Kurinskiy (Co-founder);
- Michael Voronov (Research Engineer);
- Alex Pyshnenko (Research Engineer);
- Dmitry Shakhtarin (Research Engineer);
- Anna Lekanova (Community), thesis author.

The “immediate circle of the project are advisors are people which have experience in building similar systems and variety for the project development necessary skills and network. At the moment of completing the study there are 9 advisors: Howard Wu, Emery Rose Hall, Andrey Lelikov, Lasse Clausen, Christopher Heymann, Addison Huegel, Nhan Phan, Simon Kozlov, Alexander Demidko. The main purpose of advisors’ involvement is creating and enhancing the product with the help of their recommendations, suggestions (or advises) as well as proposals for shaping the product.

The company's strategy is to create open-source protocol, broadcast or publish it to open repositories in the GitHub. The use of GitHub repositories has the following objectives:

- Involvement of stakeholders in product improvement;
- Receiving help from the interested stakeholders in the implementation of certain project tasks;
- Receiving feedback and recommendations, etc.

Summarizing, the observation data shows that advisors, as well as the external environment, the community is gathering around the core of the project. The community consists of developers who are truly interested in the product, or in a specific solution to the problem, that the product provides. Developers could either write their own code in the product repository, offer improvement or their version, creating a common shared knowledge represented as code so that other developers who want to apply the product in the future, would build their application on such commonly developed platform, could use the technology for their own projects.

To define the stakeholders for the open-source project the variety of groups was observed. The results of the study are represented in the Table 3 below.

Table 3. Fluence project Stakeholder Structure

Group	Structure
1. Internal stakeholder groups	
1.1 Projects employees	6 people (including 4 developers)
1.2 Business partners	Advisors – 9 people
2. External stakeholder groups	
2.1 Third-party (interested) developers	Global community
2.2 Future investors	
2.3 Future project representatives	
2.4 Social and public groups	

To analyze the stakeholders of the Fluence project the Mendelow's matrix was chosen. The matrix is used to assess the influence of stakeholders on the activities of the company (or project in this case), in 83.6% of organizations and is one of the most used assessment models in a variety of business communications practices. Additionally, as it was mentioned earlier, this model is relevant for researching the stakeholders of the open-source project. This model allows to evaluate the impact of each of the stakeholders on the project, depending on their level of power and interest. These indicators are evaluated on a ten-point scale, where 10 is a very strong power or stakeholder interest; 0 is absence of power or interest.

The results of applying the Mendelow's matrix to the case company Fluence are presented in the Table 4.

Table 4. The assessment of the level of interest of the key stakeholder groups of the Fluence project according to the Mendelow's power/interest matrix

Stakeholder group	Power	Level of interest	Influence = Power*Level of interest
Project employees	10	10	100
Business partners	8	9	72
Outside developers	8	8	64
Future investors	5	7	35
Future project representatives	4	8	32
Social and public groups	8	5	40

The key stakeholder groups were assessed according to open data from the internet, periodical media as well as from the following statements:

1. Project employees are engaged with the project in a greater extent; the main purpose of the participation is efficiency of the development. The “core” of the project could affect the project dynamics, its timeline and other significant characteristics.
2. Business partners or advisors vest their interest in project success, they have relatively high power because they have necessary experience for building such products. They also might have a share in the project which takes the monetary form at some point.
3. Third-party or outside developers have incentives towards the product, which could be manifested as a desire to use the product in the future, gain experience in the development of the project as well as a number of other factors driving their interest. Outside developers have the level of power above the average over the project. This is explained by the need to attract them to solve complex coding tasks and problems during the project roadmap, refinement and improvement of the project.
4. Future investors have a moderate interest in the project, since there are a lot of startups on the market. They follow project development and problem solving, identifying and resolving the bottlenecks. Future investors group has power level below the average, as they do not participate in the daily work performance of the project. At the same time, the power is not equal to zero, because when investors are included in the project, this provision could be significantly changed. Taking them into account when analyzing project stakeholders is important because they act as observers and could be actively involved in the project at any time.
5. Future project representatives’ group of stakeholders has an incentive level in the project above average, as they oversee project development, get involved the main stages of its creation, tries to understand the logic of the project, features and weaknesses. The group does not participate in active discussions, therefore, as observers, participants of this group have low power level, which will be significantly increased when a project is launched on the market.
6. Social and community groups (for example, representatives of the media or the IT industry) show moderate interest in the company's project, but at the same time, they could cause reputation damage when giving negative connotation comments, issuing opinionated materials related to the project. Therefore, the influence of these groups is rated as high.

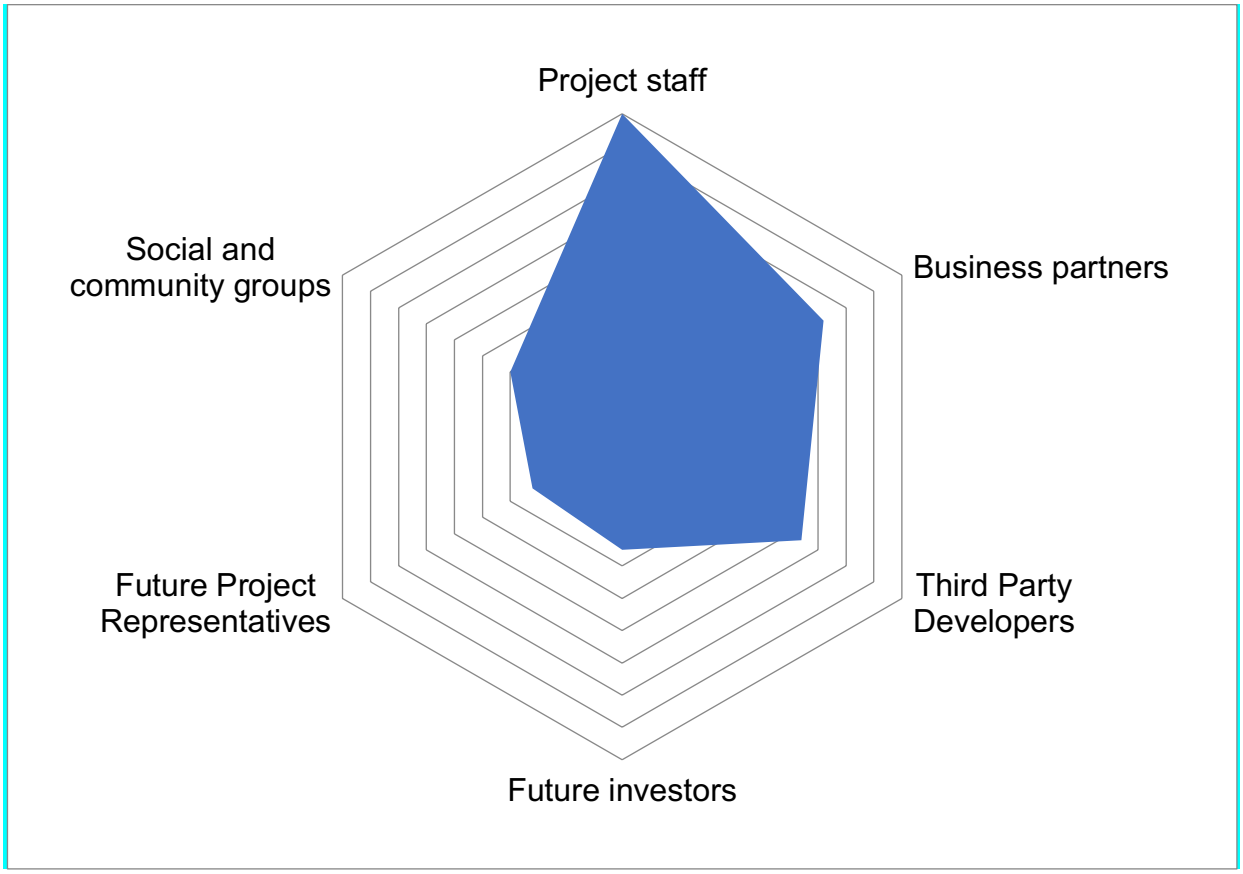


Figure 29. Stakeholder groups influence to the Fluence project according to the (adapted from Mendelow's matrix)

The influence of stakeholders on the project of the company can be assessed on the following scale:

- More than 60 points - a strong influence of stakeholders;
- From 30 to 60 points - the average influence of stakeholders;
- Less than 30 points - a weak influence of stakeholders.

The rating of the stakeholder groups of the project by their impact is shown in Figure 30.

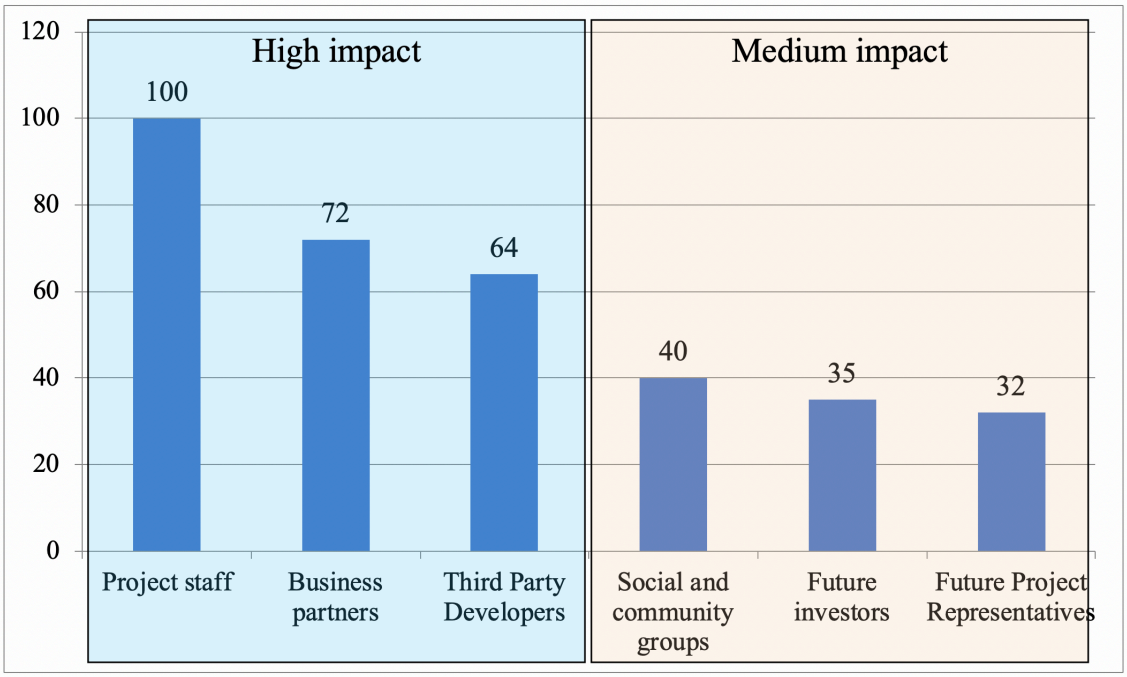


Figure 30. Stakeholder influence groups rating applied to the case project according to the Mendelow's model

Thus, not only the employees working on a project have a high degree of influence (which might be obvious), but also business partners and outside developers. The remaining groups of stakeholders have a medium impact. Groups with a low (or weak) effect were not found.

Stakeholder engagement tools used by the Fluence project team are the company's blog, as well as social media communication. These tools, in our opinion, are not effective - as they do not allow for effective feedback, moreover they do not allow for interacting with stakeholder groups, engage them in constant communication, so that they would feel more incentivized as they "own" the project, its problems, help resolving and manage bottlenecks and a set of other difficulties. The involvement of stakeholders in communication is essential to any startup.

Further the study discusses the importance of establishing community for developers and using of the proposed tools; the relevance of participants involvement in the communication. Based on the results of the additional research the stakeholder communication plan is to be elaborated which could be applied to the open-source community of developers.

4.3 Focus groups preferred form of communication in the open-source projects

This subchapter presents the findings of the qualitative and quantitative study.

Analyzing the secondary data and a number of assumptions stated in the theoretical part it was decided to conduct an empirical research to determine the specialties of the communication preferred by developers and the environment in order to form the best practice of establishing a community for an open-source project for the case company Fluence. Empirical data for this work has been collected from 1.4.2020 until 3.5.2020, the analyzing and generalizing process started from 4th of May 2020 so it could be used for the current research and forming the recommendations based on the data analyzed. Process of collecting data, the tools and methodologies used for this study were reflected earlier in the work. This chapter highlights the primary data structure.

4.3.1 Open-source project community communication analysis

This stage is structured as a qualitative study. The following research methods are used:

- Focus group research;
- Personal interview.

Focus group research.

From April 2 to April 5, 2020, at lunchtime, a series of focus group studies took place, the scenario of which was disclosed in detail earlier - in the previous chapter of this work. During the study, the following empirical data was obtained.

2.4.2020 a study was conducted in the company Fluence (4 people). All respondents believe that the community is key for an open-source project, as it allows to optimize communication between the team and the environment. Respondents emphasize that communication is the connecting link between the two groups, it allows not only to find help, but also support from the environment, is an effective “treatment for burnout”. The developers emphasize that they would like to maintain communication (interact) with the environment, which is caused not only by the practical side of the development, but also by a moral incentive - it is the community that will allow to unite everyone around the same goal, to develop and improve the product. As interaction tools, participants would prefer a branded online platform with various interaction tools, as well as meetings in real life. They highlight meetings at exhibitions, video chats, and also the collaborative accumulation of a knowledge base in an online environment as a noticeable direction.

3.4.2020, a study was conducted in the Cyberdevelopment company (4 people). Respondents also believe that the community is an important element in the development of open-source projects. They emphasize that the lack of communication with the environment, the lack of feedback tools could negatively affect the project as a whole, the motivation of developers. All respondents would like to participate in the community, receive and give valuable advice to “newcomers”, and jointly develop a product. Among the tools they would like to see online and offline forms they’d like to have more meetings or meetups in a real life (especially is they are based in the large cities), regularly interact in video chats, share experience, practices, as well as humor related to the specifics of the activity. Additionally, respondents emphasize that they do not find support in specialized groups, their circle of communication is mostly within the team, it is limited.

4.4.2020 a study was conducted at the Smart Technologies company (4 people). Respondents believe that creating a developer community is an important but difficult task. They define the difficulty in creating this community as the difficulty of holding personal meetings due to the geographical fragmentation of the teams. They emphasize that communication between them and the environment is an important but challenging task. In their responses, they emphasize that it is possible to create an online site, but without personal interaction, this site will:

- “another group on social media”, where half of the subscribers are “dead audience”;
- "all the same 3, 5, 10 people and a dozen" trolls " will participate in the communication, new information is not being received";
- “Sooner or later, the community will turn into a group, the essence of which will be the useless exchange of “GIFs” and “stickers” without any practical value for developing code, and, roughly, in a year, the group will be forgotten and will die out.”

Thus, respondents emphasize the importance of social self-identification of the group, as well as the need for personal interaction of the participants. Also, during the conversation, it was determined that the best form for the online community to exist is the form of a private club which has a strictly targeted practical aspect, excluding the access to “dead souls” and “trolls”.

5.4.2020, a study was conducted in the company Infocompass (4 people). According to the results obtained during the conversation, the importance of the community for developers was confirmed. All respondents emphasize that the community is an

important form of communication for developers, which has not only practical, but also psychological and stimulating functions. Communication in this community is the most important part, without which the community simply “will not work”. All respondents in the group want to participate in such a community, which, according to their opinion, will allow “to exchange practical experience, receive support and simply speak out.” As a form of interaction with the audience, they want to see GitHub service.

Assessing the results of the focus group study, the following research findings were made:

- The community is important for developers, as it allows to optimize communication between the team and the environment;
- The main goal of the community is to find not only help, but also support from the environment, “burnout treatment”, cohesion of like-minded people around one goal, development and improvement of the product, motivation of participants;
- Online community form: branded online club-like venue with various interaction tools, in particular with video chat, the collaborative aggregation of a knowledge base in an online environment, the involvement of experienced developers and "beginners" in the interaction;
- Offline community format, for instance, meetings in real life especially in the big cities as well as at trade fairs.

Therefore, the main characteristics of the planned community are defined.

Another area of qualitative research of the core of the project is an individual interview conducted with each of the participants. As noted earlier, the relevance of this stage is due to the fact that the individual form reveals a greater number of opinions that may not have been obtained under the influence of the «leaders» of groups (or simply – respondents were shy to bring their opinions to the discussion).

The participants of the first focus group-members of the Fluence team were the first to be interviewed. The transcript of the interview with participants of the open source project is given in Appendices 1 – 4.
The second group of respondents consisted of employees of the Cyberdevelopment team. The transcript of the interview with participants of the open source project is given in Appendices 5 – 8.
The third group of respondents consisted of employees of the Smart technologies company. The transcript of the interview with participants of the open source project is given in Appendices 9 – 12.

The transcript with the fourth group - the employees of the Infocompass company is given in Appendices 13 – 16 respectively.
The key results of the interview series are presented in Table 5.

Table 5. Key conclusions of the interview series

№	Company			
	Fluence	**Cyberdevelopment**	**Smart Technologies**	**Infocompass**
1	The possibility of feedback, multiple contacts, exchange of opinions, chat, link for which will be given to everyone	Internet community, united by a single idea with the necessary functionality, such as storage, knowledge bank, and reporting system	Something between GitHub and social community (group), where participants actively interact with each other and sometimes meet in real life	a special platform with a workspace and also a chat and forum for communication, it is possible to add an entertainment component
2	online chat system, a collaborative development tool	synthesis of various forms - interaction in the internet (it would be ideal to combine telegram, to add the ability to switch to the working workspace in it), file system with materials for development, a number of meetings in real life, participation in events, exhibitions, forums	updated GitHub, with the ability to video conferences, founding databases and similar tools, supplemented with real meetings	modified GitHub, supplemented with real-world meetings
3	online platforms, a system of text and video chats, meetings with participants,	a platform for Internet interaction on the type of social networks with a working workspace, with a video	GitHub with improved monetization of edits in open source projects	a system of online chats or channels by the type of "telegram" with switching from

	exchange of opinions and experience	communication system, as well as meetings for sharing experiences or just for spending time together		them to the workspace
4	a set of online tools (chats, forums with the possibility of voting), supported by real meetings, team-building events	any tool with the ability to get feedback, as well as periodic meetings in real life and video conferences	a special online platform like GitHub, the possibility of video meetings	a form of online platform with chats, video communication, as well as personal meetings between participants, GitHub flow or slack for free discussion

The table summarizes the main ideas of the conducted interviews.

General analysis of the results of the interview allowed us to formulate the following research conclusions. Community for all the respondents is important not only from the professional point of view, but also for the daily interaction. They see communication in the framework of community as the synthesis of online and offline loots.

4.3.2 Analysis of the stakeholder communication format within the community

The third phase of analysis is quantitative research. As noted earlier, the research of “external environment” of open-source projects in a form of online survey was also conducted to obtain valid results. The research involved 103 people. The data obtained in the course of research is presented below.

The first question of the online questionnaire was aimed at forming the sample. As a sample for this study the active participants in IT projects were chosen. Answers by the respondents with no relevant experience were eliminated.

The answers to the second question, which identifies the form in which respondents interact with the IT project community, are shown in Figure 31.

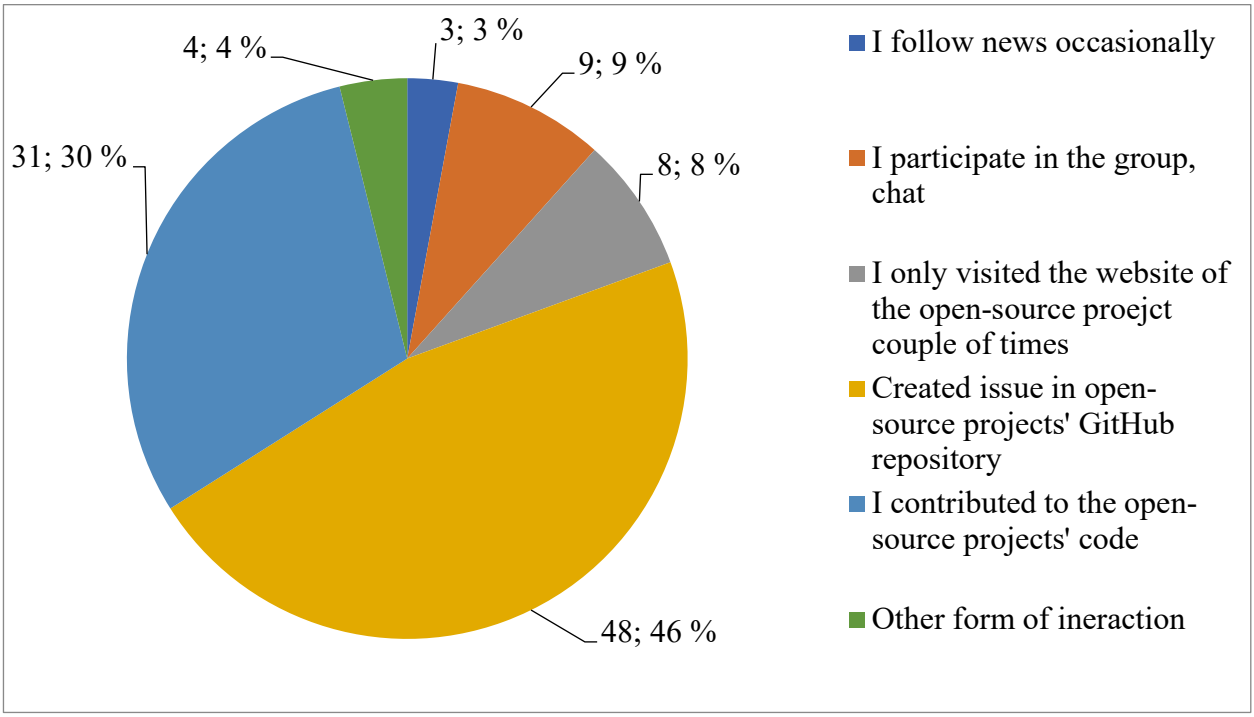


Figure 31. The format of interaction between respondents and the IT project community, people, %

Thus, evaluating the obtained data, we observe that:

1. The biggest part of project participants are issues initiators (created issue in GitHub at the core project repository or opened a new topic in the repository's “forum” with a question or suggestion) encompassing 48 respondents or 46% of responses;
2. The second largest, but not less significant, group is developers who contributed to the code, 31 respondents or 30% of the total surveyed;
3. The third largest group is represented by two subgroups: a member of the group/chat, and observers who visited the site a couple of times. These subgroups are accounted for 9 and 8 people or 9% and 8% of responses, respectively;
4. The least represented group in the survey is the followers (3 people or 3%).

Thus, open-source project stakeholders took part in the study demonstrates the proximity of the obtained data to real one, the possibility of application of this data in practice.

The answers to the third question, which reveals the role of respondents in IT projects, are illustrated in Figure 32.

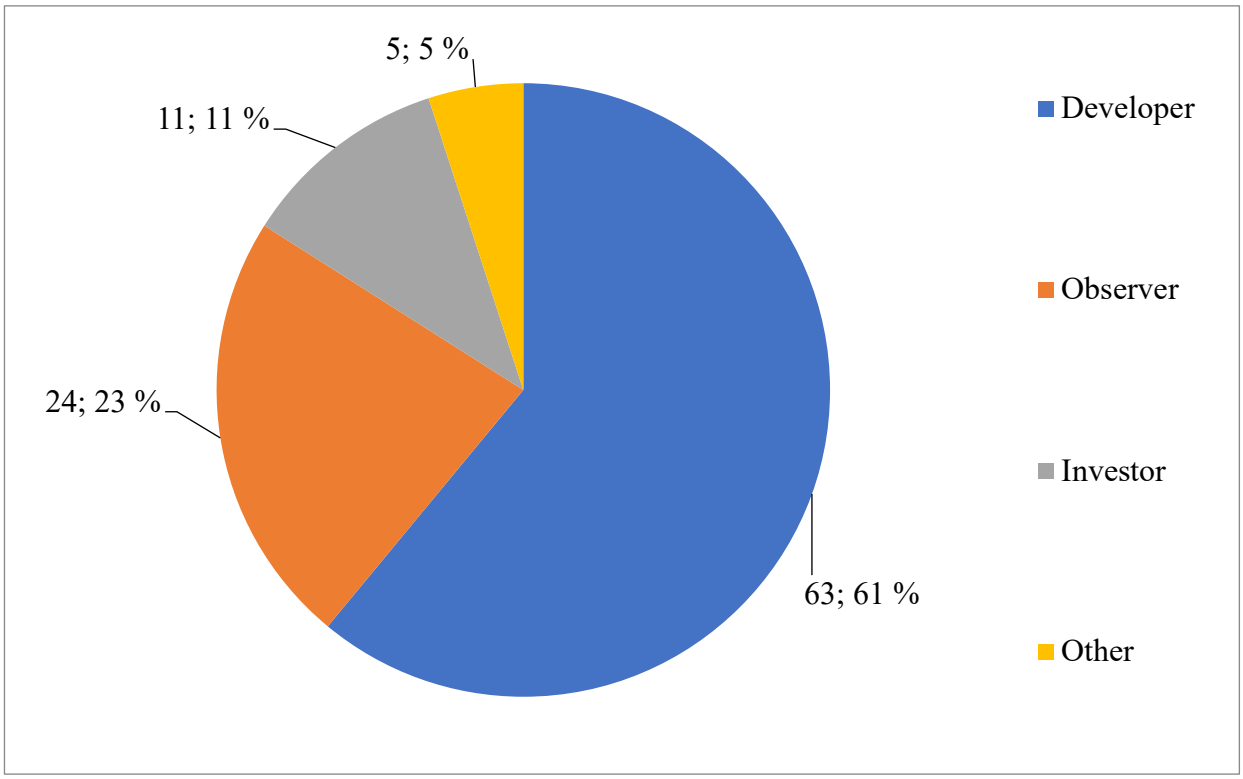


Figure 32. The role of respondents in IT projects, people; %

Analyzing the data obtained, the study shows that the majority of respondents are developers in IT projects, 63 respondents or 61% of surveyed. The second largest group is observers (24 respondents or 23% of surveyed). It should be noted that observers, since they are interested in IT projects, can also become developers in the future, and their involvement in the code development process is necessary. The least represented group in the research is investors (Figure 32).

The answers to the fourth question identify the possibility of a planned community for developers to improve the effectiveness of collaboration (or interaction) of the respondent with other participants, are presented in Figure 33.

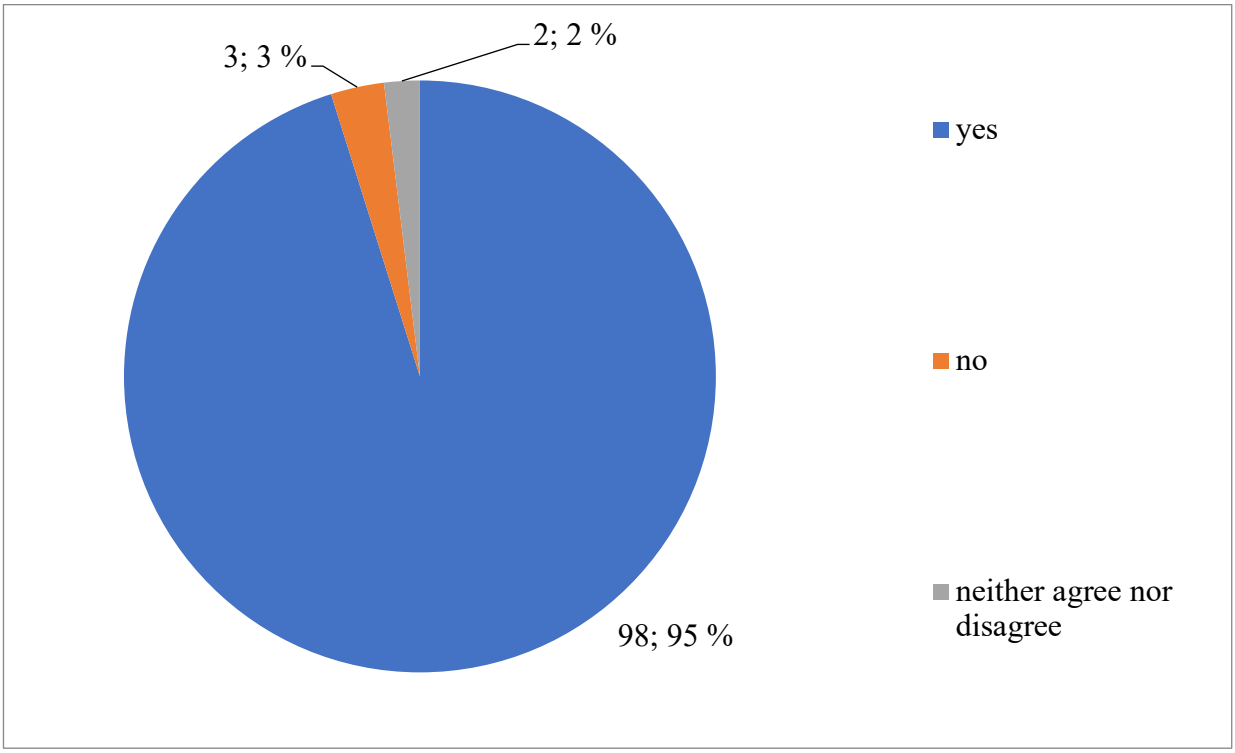


Figure 33. Do you think that establishing a community for developers would increase the efficiency of interaction with community members? People, %

Thus, the majority of respondents (98 respondents or 95%) believe that the establishment of a community will increase the effectiveness of their interaction with other project participants. Three respondents do not believe that the community affects the effectiveness of interaction between participants. It should be noted that these responses belong to investors, who, due to their workload, usually do not participate much in communication with the core developer team. Two surveyed respondents refrained from answering.

The respondents ' answers to the fifth question reveal whether communication within the IT community is important for the respondents, are presented in Figure 34.

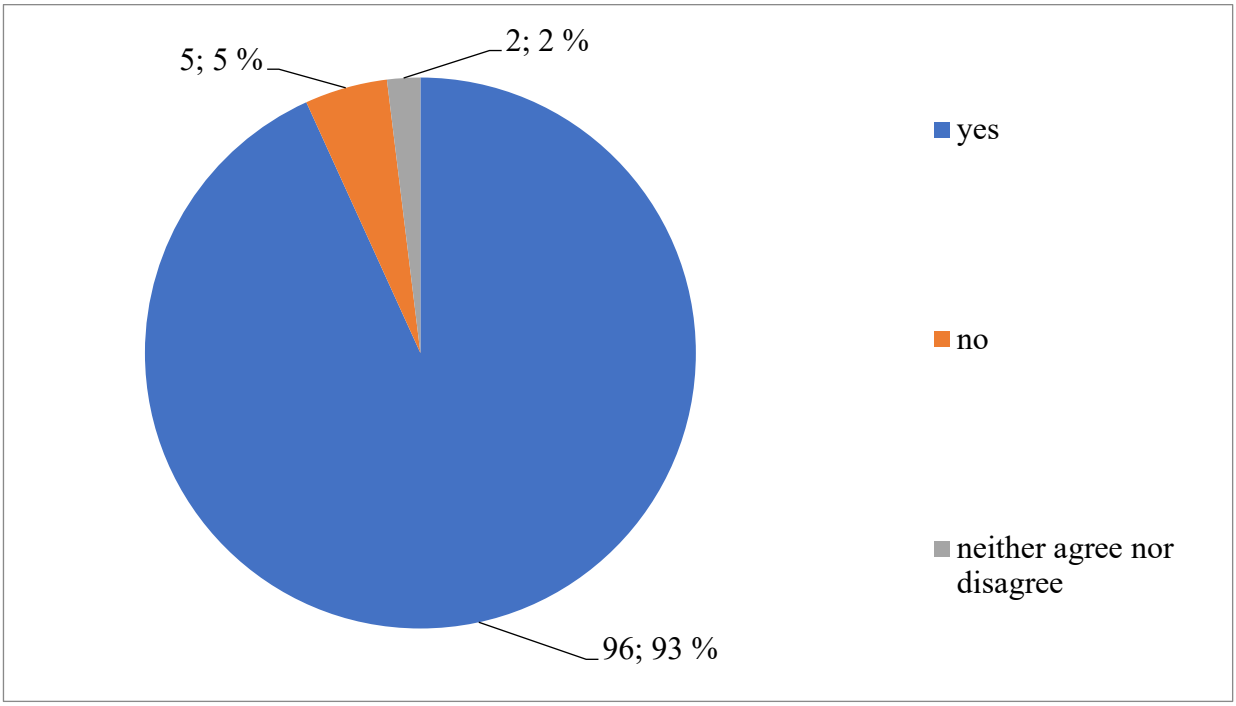


Figure 34. Do you feel that communication is necessary inside the open-source community? People, %

Therefore, assessing the data obtained, results show the communication within the IT community is important for the majority of the respondents (96 people or 93% of the respondents). Five respondents do not attach any importance to it, and 2 respondents refrained from answering.

The respondents' answers to the sixth question revealing which form of communication with other participants they would prefer when establishing an open-source IT community, are shown in Figure 35.

For a larger number of the respondents the best form of communication is any tool that provides receiving feedback, according to 35 people, or 34% of the respondents. Additionally, for a number of the respondents, the use of online chats seems ideal as emphasized by 27 people or 26% of the respondents. 22 respondents or 21% would like to see a forum with the voting option.

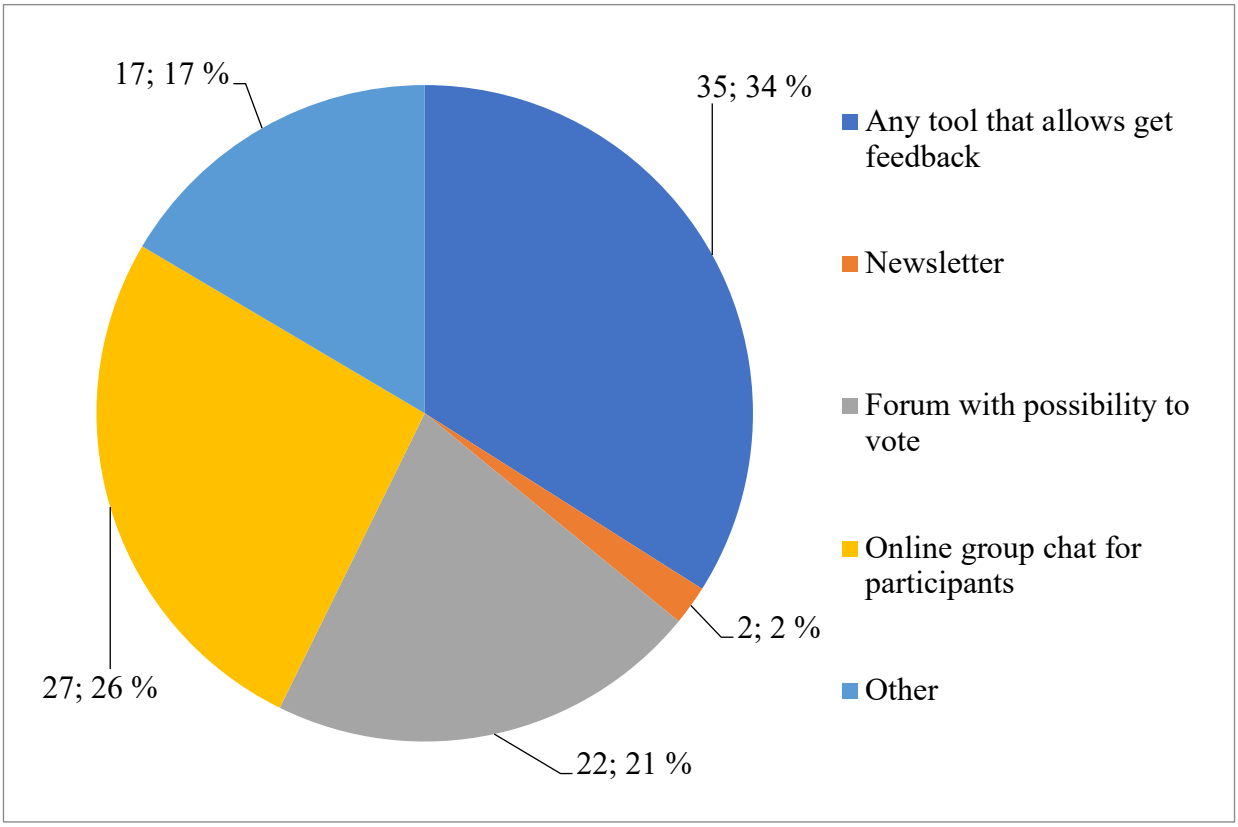


Figure 35. Which form of communication would you prefer If you took part in establishing the community for the open-source IT project? People, %

Equally relevant are the answers of the respondents who chose the category “other”. The following responses were given:

- GitHub (5 responses);
- Telegram (4 responses);
- Video conferences (3 responses);
- A simple forum (3 responses);
- All of the above (2 responses).

Only 2 respondents were interested in getting the newsletter.

The last, seventh question had an open format. It was aimed at identifying the main recommendations to establish an efficient form of communication in the open-source IT community. Despite the fact that variety of the responses was received, in different linguistic (language) forms, these responses can be standardized according to their key characteristics. In the course of standardizing the responses to the given question, the following results were obtained:

- 21 person emphasized that the use of online and offline forms is the most efficient way;

- 15 people considered the main recommendation the possibility to communicate, to have meetings;
- 12 people highlighted the need for mutual assistance and support;
- 11 people highlighted the need for events in various forms as the main recommendation. In order for the participants to meet each other, share ideas within the community such events would be a tool for finding partners to implement ideas together;
- 10 people suggested creating a knowledge base, FAQ (frequently asked questions) and visual material.

The remaining responses were presented as part of the ones already described above (for example, real communication, burnout support, task execution, etc.).

To sum up, based on the data of the quantitative research, a community is necessary for the participants of IT and open-source projects; they would like to see both offline and online forms of interaction. Analyzing the responses of the open question, study reveals that efficient community should have more than just a collaborative working environment, moreover, a place where the participants of open-source projects would be able to freely communicate not only work-related topics, but to find support, receive and provide assistance.

4.4 Summarizing the results. Choosing an effective form of communication for its further implementation in a developer community

Summarizing the results of the qualitative and quantitative research, the following conclusions were made:

- The community is important for developers and their environment;
- Communication in the community is the basis for creating strong partnerships, as well as a platform for encouragement for the developers aimed at mitigating the burnout syndrome;
- The main goal of the community is to find not only help, but also support from the environment, a “burnout treatment”; to bring together and unite like-minded people around a common goal; the development and improvement of the product, as well as motivating the participants;
- Online community should be represented in the form of a branded online platform, following the model of an exclusive or membership-only club;
- The online platform should include workshop space, storage, online chat; it should provide the possibility of creating topics at forums, joining development

(product) teams; it should contain a knowledge base, textbooks, and visual material;
- Meetings in real life in major cities, as well as at industry events (trade fairs, exhibitions), are also a necessary element of establishing the community.

These aspects were identified both during the qualitative research and the quantitative research. Therefore, the implementation requires a developer community, including the characteristics mentioned above which are represented in both online and offline forms.

Thus, the assumption that the establishing of an online community along with offline interaction (offline events) could become the main direction for stakeholder engagement in an open-source project, is validated.

5 Establishing a community for an open-source software startup

This chapter discusses the structure for the plan of the community for an open-source project. The subchapters discover practical recommendations for establishing a community along with the detailed proposal of an online community platform, offline events plan followed by applied guidelines for each stage of the community development.

Based on the results the structure of the planned community was elaborated as well as key recommendations of using the structure provided. Significantly, the results obtained are correlated with theoretical research which discusses the two communication forms relevant for the open-source communities: online along with offline. Simultaneously as stated earlier in the work to overcome the chasm of the project lifecycle it is suggested to involve offline events immediately which could serve as a basis to community engagement.

5.1 Designing the structure for the developer community

In order to increase the communication efficiency of the Fluence project and engage stakeholders in communication, it was proposed to establish a community of developers which will facilitate the launch an open-source software. An upcoming community should include the following tools and activities:

- An online platform for the interaction between the core team and the stakeholders of the project;
- An internal Code of Conduct or community engagement code;
- Offline meetings such as events, meetups during the technological industry congresses, summits, and other bigger IT events (for example, Berlin Blockchain week, SF Developer week, etc.);
- A system of video chat and personal chats aimed at the interaction between all participants, encouraging communication and a collaborative spirit;
- Organizing online conferences to encourage communication, meeting new participants;
- Documenting the stages of compilation, merging, launching, testing and deploying the code by using a simple web tool out-of-the-box;
- Development of tutorials, videos, documentation and helpful visual materials for outside or third-party developers who want to use the product, to contribute to code or build on top of the OS protocol;

- Maintaining publicly available roadmaps, timelines with upcoming changes, forks, implementing or supplementing new features and functions;
- Interactive event planning.

In the framework of the above-mentioned tools and areas, an establishment of an online community which includes all of the above tools is proposed. It should be noted here that the online community (which is proposed to be the basis for communication among participants) is not a public group in social media. A distinctive feature of the proposed interaction is the creation of a dense network of relations between participants – project stakeholders. The proposed structure of the community of Fluence project developers and stakeholders community is presented on Figure 36.

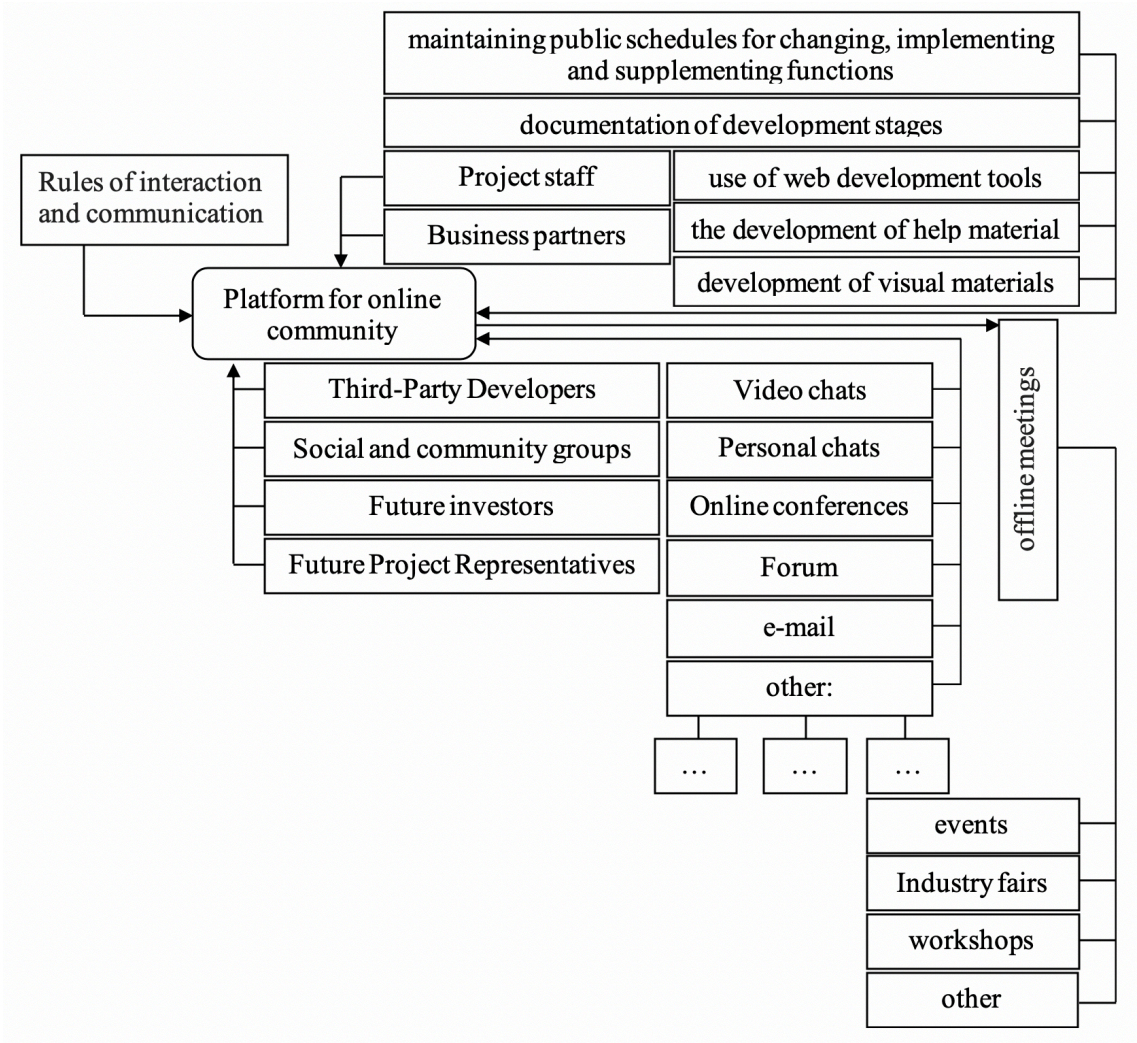


Figure 36. Developer community structure

The basis for the development of the online community - the main platform for communication among stakeholders, should be documentation support. The main document within the community is proposed to be "README" – a kind of community code, including:

- Description of the project;
- A description of core team of the project;
- A description of goals and objectives;
- The relevance of project development;
- Mission and community values.

After developing a community code, it is important to develop related documentation, such as:

- Documenting the stages of compilation, merging, launching, testing and deploying the code by using a simple, tool-free web tool;
- Development of tutorials, videos, guides and visual materials for working with code;
- Maintaining publicly available roadmaps, implementing and supplementing functions;
- Interaction events planning.

In these areas, the nature of the community, its mission and values should be pursued, preferably using a trademark or brand (Figure 37.)

Figure 37. Trademark (logo) of the project Fluence (adapted from Fluence Labs)

It must be emphasized that the use of the trademark in the materials and in the design of the community helps participants psychologically feel part of the team, which affects their commitment and involvement.

The developed documentation should be posted on the online platform for communication. The online form was a rational choice because the Fluence team is decentralized, i.e. the participants are geographically distributed. The territorial or geographical distribution of the members is one of the features of open-source software development teams and is also characteristic for most of the startups. Undoubtedly, the members in this community will also represent different parts of the world and be distributed worldwide. The structure of the proposed platform for the developer community is illustrated on Figure 38.

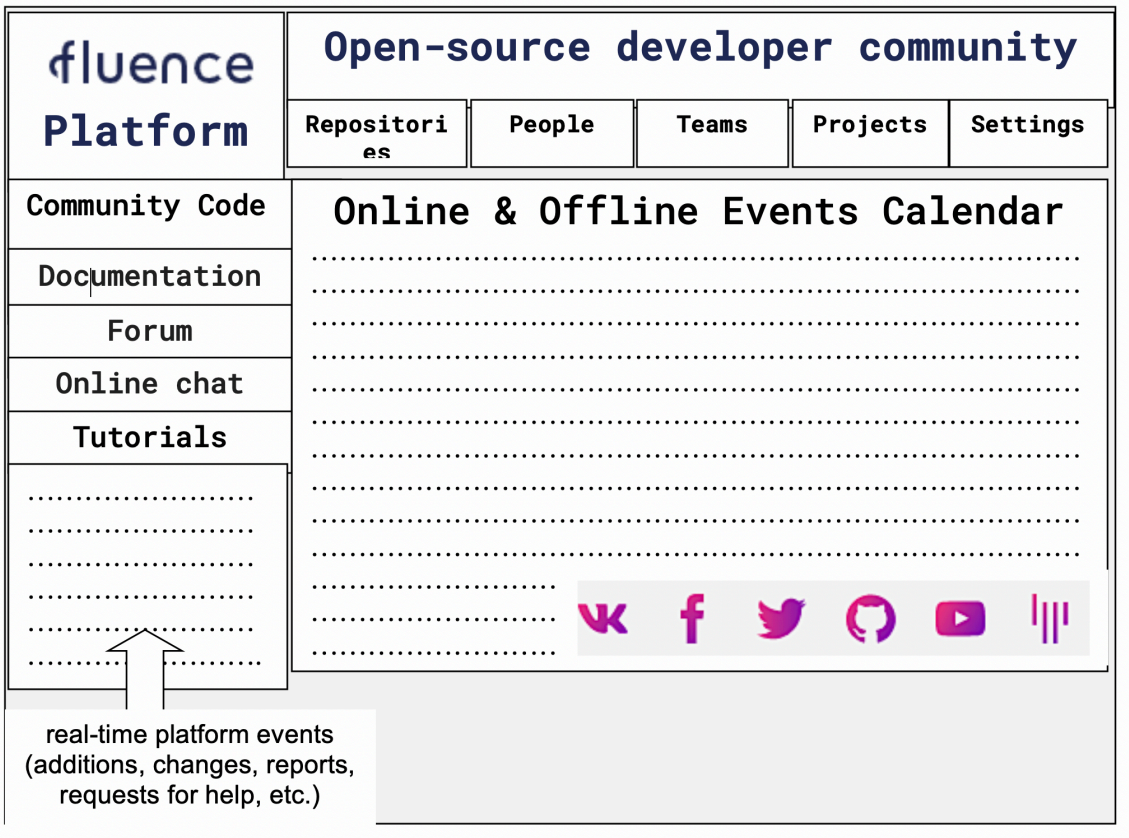


Figure 38. Proposed open-source development community platform

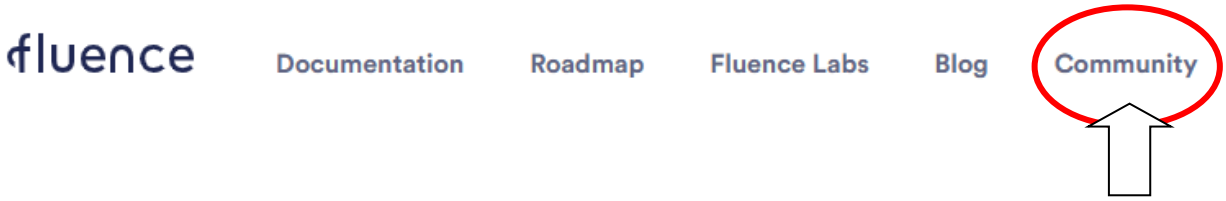
This platform has a wider focus than just code development. In addition the platform for such community sparks the interaction between participants united by a common idea of developing a decentralized web, as well as involving participants who are triggered by the same values, students who want to gain development experience, as well as investors and future representatives of the company, who want to follow the product from the beginning of its appearing on the market e.g. from the stage of development. The proposed online platform helps solving a number of problems of practical importance:

- Helps formation of the groups of participants for certain tasks or projects;
- A list of teams and issues/projects they are working on;
- A forum where everyone can create their own topic and receive help or support;
- An online chat to solve urgent issues with participants who are currently online or ask a question from the core team.

The forum depicts a number of urgent tasks necessary for solving within the project, divided by user level. A number of tasks will also be given to engage students and beginners in programming. As noted earlier, the platform will feature educational materials, knowledge base, and graphic diagrams.

The interactive window of additions would be useful for tracking changes, reports, requests for help, etc. It should broadcast decisions of changes in the product, the top of topics on the forum, and also welcome new members of the community (by the pop-up messages: "New member just joined us. Welcome, username. Say hi!"

The most significant events of the platform are proposed to be broadcasted on social media (to a lesser extent), but at the same time sufficient to attract new participants. The place of the community of the Fluence website structure is as illustrated by Figure 39.



The place of the online community in the general form of the site

Figure 39. The placement of the “Community” tab in the general Fluence website

These findings support the notion that creation of an effective community based only on one of the online instances is impossible. In order to provide the full interaction and feedback between the stakeholder groups we propose the online and offline events calendar as one of the central elements of the platform. This suggests that the following should be submitted:

- Offline meetings calendar (joint events, meetings at exhibitions, etc.);
- Video interaction of the "core" and the environment;
- Workshops, meetups and online conferences for everyone.

These events with the visible schedule should be presented in the appropriate platform window and duplicated in own social media channels of the project. The proposed online community should be deployed in private access mode and act as a private club of participants. This is an important finding in the understanding of the nature of the communication in the big groups. Creating an online platform for communication is necessary so that people can freely communicate with each other, to turn them from a crowd of users into true supporters and followers who are actively developing the project, contributing to the development and improvement of the code.

It should be noted that it is necessary to establish communication between developers and other stakeholders for the community to start functioning. This interaction process is one of the most important in launching any community.

5.2. Guidelines for the stages of community establishment

The main goal of the establishing community is to introduce users to the interaction, keeping them in the structure of the community. Moreover, turn community members with minimal activity to developers which are actively assisting to the "core" and contributing to the code. Figure 40 shows the funnel for engaging community users.

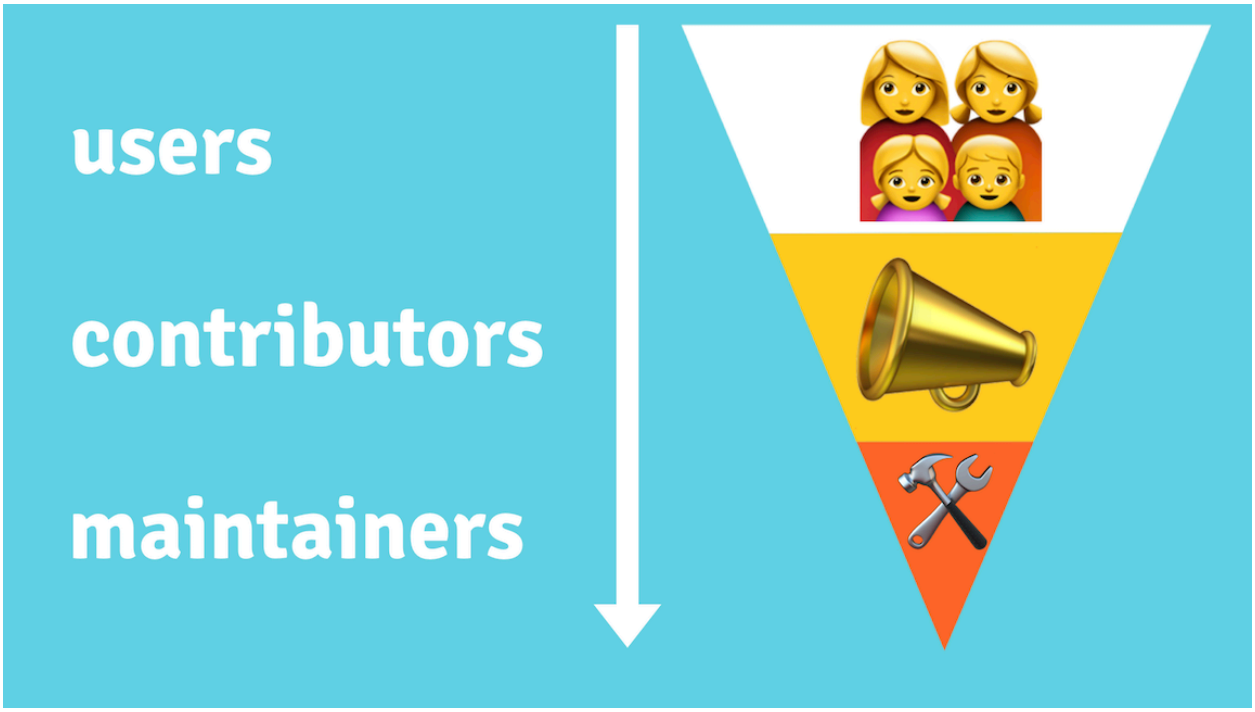


Figure 40. Community members funnel (adapted from Mike McQuaid, opensource.guide)

The setup is considered to be generic, however previous studies have shown the rule "90/9/1" works for most of the interactions of the organization (Nielsen, 2006):

- 90% are ordinary participants;
- 9% are commentators;
- 1% are participants ‘useful’ to the community, participating both in the creation of content, surveys and discussions, and participating in the development and improvement of the code.

The basic efficient community characteristics should be kept such as:

- Responsiveness: prompt response to questions, comments, as well as topics on the forum;
- Mandatory, automatic and manual online greeting of each participant;
- Three types of agendas (applied, external and internal agendas).

The recommended proposal for the communication within the community as follows:

Stage 1: attracting members. To date, the community is represented by a core team (6 people), as well as advisors (9 people), the number of other participants (previously considered external stakeholders) is extremely small, their main characteristics, as well as participation drivers are not known. At this stage, it is recommended to research people talking about the project on social media and forums, find people attracted by the same values and mission (e.g. project) and start working with them. Targeted actions to attract stakeholders to the community, the activating of the network will help form the initial core of loyal people, which will become the basis of the community. The following areas will be used here to engage and bring people together:

- Forums and content are to be initiated by company representatives to engage participants;
- Creation of common projects both online and offline;
- The presence of a dense internal graph (network) using video communication, congratulations, various team-building online events.

The main focus of creating a community at this stage is quality, not the number of participants to create a stable network of the top-notch developers who would attract less experienced developers in the future. It has been proven by practice of the author that it is better to have loyal members engaged with the project than inactive members pushing others away from the community.
A remarkable way to attract new members could be to use the built-in buttons of social media "Share", so that users more actively talk about the community.

Stage 2: involvement of members in communication. After passing the first stage of gathering participants, traditionally, there coming a "stage of awkward silence." In the framework of this stage of the community emergence, it is necessary to use the following tools to engage participants:

- Involvement of participants through mentions;
- The creation of provocative themes;
- Activation of inactive participants using a personal (auto-distribution) invitation to the conversation.

Involvement of participants in communication will create close partnerships in the community, the desire to communicate more often. At this stage, it is necessary to organize meetings, events, gather at exhibitions. This will allow participants to recognize each other not only by profile photo and voice in video chats, but also introduce them to each other personally. A suggested area here can be meetups, events or meetings at major exhibitions and in large cities. A method of determining the location can be used by creating a public survey within the online community to raise the engagement level. Analyzing the planned events for the end of the year 2020 – beginning of 2021 year, the following event calendar was proposed (Table 6).

Table 6. Meetups with community members for the end of 2020 – beginning of 2021.

Meetup	Date	Title
Helsinki	11.06.20	Building Decentralized Networks
Tallinn	27.07.20	Fluence / Polkadot Meetup
Berlin	10.09.20	Decentralized Stack. Developers Meetup.
Berlin Closing Party	11.09.20	Berlin Blockchain Week Closing Party
Barcelona	18.09.20	Building a Decentralized Data Infrastructure
Warsaw	21.09.20	Building a Decentralized Data Infrastructure
SF	08.10.20	Web3 stack, what's next for developers?
Berlin Workshop	24.10.20	A Query Layer for Dapps
Prague meetup	30.10.20	Web3 stack, what's next for developers?
Paris Dapp Meetup	05.03.21	Understanding Dapp Ecosystem, a sneak peak into the future
Paris Fluence workshop	06.03.21	Building Decentralized Backends
Paris Web3 Stack	07.03.21	IPFS, Fluence, NuCypher. Web3 stack, what's next for developers?

This list is to be added according to the planning of the new meetups and other events as the situation in the world might change. It is important to note that this study is partly being completed during the current pandemic COVID-19 which restricts movement worldwide as well as holding events. The future locations will be chosen according to the new governments' guidelines.

Stage 3: turning commentators into partners. After involving commentators in the interaction, we suggest creating a loyal community out of them. We could also call this

stage “the stage of task delegation”. Within the framework of this phase, the “projects” and “repositories” section will host the problematic aspects of the code, and recruit teams from among the community members who will be able to refine and improve the code. Moreover, the open-source section will always be open for changes.

It should be further noted that establishing a community is an extremely labour-intensive and longstanding process based on the daily project representative (community manager) work of getting acquainted and meeting with the participants – both online and offline. Personal acquaintance with each of them will create a solid foundation for the further development of the community. An active community will help attract new users, maintain the interest of existing ones and provide valuable feedback for the core team that will help improve the product.

In conclusion, it should be emphasized that in order to create an effective startup project community in general, or OS in particular, the first subject to look at is a care for people, and after that a care for their contribution. Naturally, the beginner developers’ contributions are not as noticeable as of experienced participants, so it is tremendously vital that care for people and their passion for the project should be higher than need for correcting mistakes. If the right people come together around the idea, the correctness and speed of development as well as improvement of the code will be certainly achieved. Not a single startup OS code can survive without a community that cares.

As a result, the study proposes to build communication in the community using the following tools:

- Online platforms;
- Interaction with participants using online tools;
- Establishing an offline meeting system.

6 Conclusions

This chapter describes the process of working on the research in the context of reliability and validity, as well as provide an assessment of own findings and learnings according to the author. At the end of the chapter the limitations of the research and further suggestions on the topic of work are given.

6.1 Reliability and validity of the research

In this subsection, the process of writing the work is assessed in terms of validity, reliability and
credibility.

Validity is a significant factor when carrying out constructive research. Validity is divided by internal, credibility, and external, transferability. Often the qualitative method is lacking validity and researcher should involve respondents in checkup processes to ensure and get more credibility for the results. The data validity relations were overseen throughout the entire research, from the stage of planning until the findings phase. However, qualitative and quantitative studies do not provide absolute truths or facts concerning reliability and validity, which is explained by the user activity which positively affected the case company. The qualitative data collection, interviews, were duplicated to four different companies; quantitative data collection, the survey, was spread among different channels to get an unbiased sample in order to obtain reliable and valid data. The importance of validity should be also considered in order to make the research transferable. To contribute to the common knowledge the model of data collection as well as analyzing the results are clearly presented in this work and represent an applied pattern thinking, making the entire research a model or pattern for further use.

Reliability shows how much are the results trustable in correlations with methods that have been used. Reliability measures the extent to which research results can be replicated.

The current work has collected a set of tools described in the empirical part of the research which could be compiled and applied to another research of the community. According to an analysis of theoretical sources, primary and secondary data, this degree can be defined as high. The results were presented to the Fluence Project team and were approved to be implemented. Therefore, the obtained data is in demand and useful.

During a number of conversations with the company representatives it was found out that they also discussed the need to create a community and raised questions about its effectiveness. Therefore, the results can be assessed as credible.

All in all, the validity and reliability of this study is confirmed by the presented arguments, as well as by the case company team. The research results are compatible and applicable to the theoretical base, provide the answers to the questions and obtain the assumed results.

6.2 Reflections on learning

It stands for the reason that the named topic was chosen by the researcher. Currently the author works as a community manager at the Fluence open-source software company. In the process of working for the company, the author observed problems in the complexity of attracting members to grow the community. Moreover, there was a tendency for members to quickly drop out of the project. This all necessitated the development of such a tool that could keep participants in the project, additionally motivating them to be more active, as well as constantly attract new members.
The basis obtained during the university studying provided the formation of theoretical knowledge supported by its practical application in the process of community development.

The results obtained during the empirical study confirmed the assumptions kept by the author, on the basis of which the mix of theories proposed for the establishment of the community for the open-source project can be considered successful for the community management practices.

The key challenge in writing the work was the process of developing research tools such as focus group research script basis, interview questions, and surveys furthermore the search of the leads along with the collection of the results had to be well planned greatly in advance.

Upon completion of the final practical chapter on the community establishment, the proposed project was submitted to the Fluence team for the review. According to the data obtained, it was revealed that the company has a willingness to apply the results in practice and appreciates the quality of the study.

This research facilitated the experience on the practical part of this work, conducting focus group interviews and surveys, moreover, the research helped obtain the experience of determining finding patterns in theory. The literature review generated extensive knowledge of the variety of theories, in particular, the theory of organizational development, which is clearly traced in the development of open-source projects, stakeholder theory has opened an interesting angle to this work.

Another area of interest was the study of the profile of a typical IT employee. Despite the fact that there is very little research in this area, the study helped to overcome stereotypes in this space for the future. A software developer is no longer a typical introvert, but rather a person who needs to communicate, though, communicate mostly in the IT circles due to the special way of thinking, slang and other factors demonstrating professional transformation.

The study supported gaining learning from the literature review in addition to the practical experience in conducting a comprehensive study, which makes it credible to offer directions for optimizing communication in the developer community, as well as specific activities for the management of the community. Due to this research, it was possible to contribute to the creation of an open-source community initiative for a decentralized storage project, which allows users for more secure storing information and in the future will provide the possibility to store viable data decentrally without losing its confidentiality, quality and volume. Attracting participants to create this open-source project within the community will hopefully affect the development of the open-source movement as a whole.

6.3 Limitations of the study and recommendations for further research

The key limitation of this study was the limited amount of time that the author had to use in addition to working time. If it was possible to get a study leave for several months, the process could be more effective. In addition, the number of respondents participating in the survey was rather small, but, nevertheless, a sample was presented for all groups of stakeholders of the company.

Despite the fact that the proposed community structure is currently being introduced (an online platform for communication with stakeholders is being developed), this direction was poorly studied in the previous research thus could also be further developed.

The directions for further research are proposed as following:

- The process of activation the passive community members;
- The process of transformation of members into the advocates and ambassadors of the project;
- The development of an internal community brand.

In general, any community should develop along the plan-do-check-act cycle by W. Edwards Deming (Kadir at al, 2019), that is, constantly enhance and refine.

References

Aclu.org. NSA Surveillance. Available at: https://web.archive.org/web/20200226112535/https://www.aclu.org/issues/national-security/privacy-and-surveillance/nsa-surveillance. Accessed: 5 May 2020.

Anderson, C. 2005. A qualitative study of the effectiveness of a leadership development program on community sustainability and the activities that influence community improvement in the context of sustainable development. Available at: https://haaga-helia.finna.fi/PrimoRecord/pci.proquest305342346. Accessed: 12 May 2020.

Bacon, J. 2018. The Art of Community. O'Reilly Media.

Barrow, B. 2017. Stakeholder Management: 50 Ways That you can Become Brilliant at Project Stakeholder Management, or How to Engage, Inspire and Manage Even Difficult Stakeholders. CreateSpace Independent Publishing Platform.

Baugh, A. 2015. Stakeholder Engagement: The Game Changer for Program Management. Auerbach Publications. Boca Raton.

Bellini, A. 2019. Open Source: It's Not That Scary. Tech Advisor, 6, pp. 19 – 26.

Block, P. 2018. Community: The Structure of Belonging. Berrett-Koehler Publishers. Oakland.

Brandford, L. 2018. Why we need to talk about burnout in the tech industry. Forbes.com. Available at: https://www.forbes.com/sites/laurencebradford/2018/06/19/why-we-need-to-talk-about-burnout-in-the-tech-industry/#59e9a1971406. Accessed: 1 May 2020.

Brasseur, V.M. 2018. Forge Your Future with Open Source: Build Your Skills. Build Your Network. Build the Future of Technology. Pragmatic Bookshelf. London.

Chydenius, T. & Gaisch, M. 2016. Work-life Interaction Skills: An Exploration of Definitional and Functional Perspectives within the Austrian and Finnish ICT Industry. Business Perspectives and Research, 4(2), pp. 169-181. Available at: https://haaga-helia.finna.fi/PrimoRecord/pci.sage_s10_1177_2278533716642654. Accessed: 16 May 2020.

Clark, K. March 2019. Y Combinator president Sam Altman is stepping down amid a series of changes at the accelerator. Available at: https://techcrunch.com/2019/03/08/y-

combinator-president-sam-altman-is-stepping-down-amid-a-series-of-changes-at-the-accelerator/. Accessed: 4 March 2020.

Chasinga, J. 2020. Dealing with programmer's burnout. Innovation & Tech Today, 3, pp. 22 – 26.

Clayton M. 2018. Stakeholder Engagement: Advanced Insights to the Essential Discipline for Mature Project Managers. O'Reilly Media. California

Constable, G., Fishburne T., Rimalovski F. 2014. Talking to Humans: Success starts with understanding your customers. O'Reilly Media. California.

Cornelissen, J. 2014. Corporate Communication. A Guide to Theory and Practice. 4th ed. SAGE Publications. London.

DiBona, C., Stone M., Cooper D. 2008. Open Sources 2.0: The Continuing Evolution. O'Reilly Media. California.

Dixon, C. 2018. Why Decentralization Matters. https://onezero.medium.com/why-decentralization-matters-5e3f79f7638e. Accessed: 7 March 2019.

Dobni, C.B., Luffman, G. June 2003. Determining the scope and impact of market orientation profiles on strategy implementation and performance. Stratagic Management Journal. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1002/smj.322. Accessed: 21 April 2020.

Effenberge, F. 2018. What the open source community means to me. Solid State Technology Magazine, 11, pp. 14 – 22.

Fogel, K. 2005 Producing Open Source Software: How to Run a Successful Free Software Project. O'Reilly Media. California.

Freeman, R. 2010. Strategic management: a stakeholder approach. Cambridge University Press. New York.

Freeman, R., Harrison, J., Barney, J., Phillips R. 2018. The Cambridge Handbook of Stakeholder Theory. Cambridge University Press. Cambridge.

Fluence Labs, January 2019. Dapp Survey Results. Available at: https://medium.com/fluence-network/dapp-survey-results-2019-a04373db6452. Accessed: 24 January 2020.

Gamalielsson, J. 2014. Sustainability of Open Source software communities beyond a fork: How and why has the LibreOffice project evolved. Journal of Systems and Software, 89, pp. 128-145.

Gray, T. 1998. Code of conduct. Business, North Carolina, 18(4), p. 32.

Haff, G. 2018. How Open Source Ate Software: Understand the Open Source Movement and So Much More. Apress. New York

Hintjens, P. 2018. Emotional burnout of volunteers. Available at: https://habr.com/ru/company/philtech/blog/345982/ Accessed: 2 April 2020.

Hintjens, P. 2016. Social Architecture: Building On-line Communities. O'Reilly Media. California.

Holland, V. 2015. Challenges help developers get started with other projects. Technology News, Gadgets and Reviews, 8, pp. 72 – 79.

Hu, D., Zhao, J. L. & Cheng, J. 2012. Reputation management in an open source developer social network: An empirical study on determinants of positive evaluations. Decision Support Systems, 53(3), pp. 526-533. Available at: https://haaga-helia.finna.fi/PrimoRecord/pci.elsevier_sdoi_10_1016_j_dss_2012_02_005. Accessed: 7 April 2020.

Iriberri, A, Leroy, G. 2009. A life-cycle perspective on online community success. ACM Comput Surv 41(2):1–29 Available at: http://web.b.ebscohost.com.ezproxy.haaga-helia.fi:2048/ehost/pdfviewer/pdfviewer?vid=1&sid=3a383fb4-8000-4817-ad4e-102d1eef3eb7%40pdc-v-sessmgr06. Accessed: 13 May 2020

Kadir, B., Broberg, O., Carolina S., Jensen, N. 2019. A Framework for Designing Work Systems in Industry 4.0. Available at: https://www.cambridge.org/core/journals/proceedings-of-the-international-conference-on-engineering-design/article/framework-for-designing-work-systems-in-industry-40/FFF8F6DD0C3269B48E102BA2708A3CC4. Accessed: 15 May 2020.

Karakulov, M. 2019. Professional burnout in the IT field. HABR. URL: https://habr.com/ru/company/habr_career/blog/437264. Accessed: 24 January 2020.

Kraut, R.E., Resnick, P. 2012. Building Successful Online Communities: Evidence-Based Social Design. The MIT Press. Cambridge

Lindberg, V. 2008. Intellectual Property and Open Source: A Practical Guide to Protecting Code. O'Reilly Media. California

Lukka, K. 2003. The constructive research approach. p.83.

Magnani, L., Carnielli, W., Pizzi, C. 2010. Model-Based Reasoning in Science and Technology.

Manjoo, F. October 2017. Interview for the Fresh Air podcast via npr.org. Available at: https://www.npr.org/2017/10/26/560136311/how-5-tech-giants-have-become-more-like-governments-than-companies. Accessed: 3 May 2020.

Marin, A., Mitchell, R. K., Lee, J. H. 2015. The vulnerability and strength duality in ethnic business: A model of stakeholder salience and social capital: JBE JBE. Journal of Business Ethics, 130(2), 271-289. Available at: http://dx.doi.org.ezproxy.haaga-helia.fi:2048/10.1007/s10551-014-2207-7. Accessed: 18 May 2020.

Martin, G. 2015. 4 tips for breaking into an open source community. Innovation & Tech Today, 12, pp. 56 – 61.

Martinez, K. 27 March 2020. 7 strategies for running effective remote meetings. Available at: https://zapier.com/blog/effective-remote-meetings/. Accessed: 17 April 2020.

Masson, P. 2015. 17 years of defending open source: Join the OSI today. Tech Advisor, 6, pp. 17 – 29.

McQuaid, M. Building Welcoming communities. Contributor funnel. Available at: https://opensource.guide/building-community/. Accessed: 15.4.2020.

Mendelow, A. L. 1991. Stakeholder mapping. Environmental Scanning: the impact of the stakeholder concept, proceedings from the second international conference of information systems, pp. 407 – 18. Cambridge, MA.

Miles, J.A. 2012. Management and Organization Theory. Jossey-Bass. San Francisco.

Millington, R. 2012. Buzzing Communities. Feverbee. Vermont.

Mitchell, R.K., Agle, B.R, Wood, D. J. 1997. Toward a theory of stakeholder identification and salience: defining the principle of who and what really counts. Academy of Management Review 1997, Vol. 22, No. 4 835 – 886.

Moreno, M. 2006. Open Source: A Multidisciplinary Approach. World Scientific. Republic of Singapore.

Newbould, G., Luffman, G. 1989. Successful Business Politics. London: Gower. pp. 78 – 79

Nielsen, J. 2006. The 90-9-1 Rule for Participation Inequality in Social Media and Online Communities. Available at: https://www.nngroup.com/articles/participation-inequality/. Accessed: 14 April 2020.

Oram, A., Bhorat, Z. 2018. Open Source in the Enterprise. O'Reilly Media. California.

Oyegoke, A. 2011. The constructive research approach in project management research", International Journal of Managing Projects in Business, Vol. 4 Iss: 4, pp.573 – 595. Available at: http://www.emeraldinsight.com/action/doSearch?ContribStored=Oyegoke,+A, Accessed: 5 June 2019.

Pazderam, R. 2015. A developer's guide to getting into open source. E&T Magazine, 1 (3), pp. 6 – 11.

Peatfield, N. 2015. 5 open source developers tell us how they got started. MIT Technology Review, 7, pp. 17 – 24.

Pearce, J. M. 2018. Open-source hardware could defend against the next generation of hacking. WIRED Magazine, 5 (12), pp. 37 – 44.

Raymond, E.S. 2001. The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary. O'Reilly Media. California.

Rowley, T. 1997. Moving beyond dyadic ties: A network theory of stakeholder influences. Academy of Management. The Academy of Management Review, 22(4), pp. 887-910. doi:10.2307/259248. Available at: https://search-proquest-com.ezproxy.haaga-helia.fi/docview/210943667?pq-origsite=primo. Accessed: 24 May 2020

Schmitz, E. A., Baum, M., Huett, P. & Kabst, R. (2019). The Contextual Role of Regulatory Stakeholder Pressure in Proactive Environmental Strategies: An Empirical Test of Competing Theoretical Perspectives. Organization & Environment, 32(3), pp. 281-308. Available at: https://haaga-

helia.finna.fi/PrimoRecord/pci.sage_s10_1177_1086026617745992. Accessed: 19 April 2020.

Simonite, T. May 2018. The Decentralized Internet Is Here, With Some Glitches. Wired. Available at: https://www.wired.com/story/the-decentralized-internet-is-here-with-some-glitches/. Accessed: 12 May 2020.

Smelt, P., Staves M. 2019. Stakeholder Management: What you need to know: A 6 Point plan for success. Something To Say Publications. Ontario.

Smith, D. 2015. Open source licensing practitioners speak on today's issues. Network Computing, 6 (12), pp. 32 – 41.

Tamburri, D. A., Palomba, F., Serebrenik, A., Zaidman, A. 2018. Discovering community patterns in open-source: a systematic approach and its evaluation. Empirical Software Engineering, 11, pp. 21 – 38.

Thompson, J.L. 2001. Understanding Corporate Strategy. p. 610. Cengage Learning EMEA. Available at: https://books.google.ru/books?id=WZfoXlgTWzgC&lpg=PA233&ots=KG1d1DODZ_&dq=mendelow%201991%20Proceedings%20of%202nd%20International%20Conference&hl=ru&pg=PP6#v=onepage&q&f=false. Accessed: 30 May 2020.

Whitehurst, J. 2015. The Open Organization. Harvard Business Review Press. Cambridge.

Volpi, M. 2019. How open-source software took over the world. Computer World, 6, pp. 17 – 22.